MASTER 2 Sciences Du Langage Parcours langues, langages et enjeux sociétaux

Rapport de stage long

MESURE DE L'ECART DU GENRE DANS LES OFFRES D'EMPLOI D'INDEED CANADA

Etude quantitative et qualitative

Effectué par :
Fella Bennadji

Sous la direction de :
Marc Allassonnière-Tang
Florence Chenu
Rémi Anselme

Septembre 2022

Mon séjour de six mois au laboratoire DDL fût une première et belle expérience dans un laboratoire de recherche. Cette expérience a été guidée par mon directeur de recherche et tuteur de stage Marc Allassonnière-Tang qui a, avec beaucoup d'attention et de patience, suivi mes premiers pas en TAL. Je le remercie infiniment pour sa disponibilité, ses conseils et ses orientations.

D'énormes remerciements s'adressent à ma co-directrice de recherche Florence Chenu et à Rémi Anselme, doctorant au DDL, pour son précieux aide et explications.

Enfin je remercie les membres du laboratoire DDL pour leur accueil et leur bienveillance, en particulier Matthew Stave, Lucie Métral et Karl Seifen

MERCI !

TABLE DES MATIERES

Résumé scientifique

Au Canada, un pays connu pour être très soucieux d'équité et de parité professionnelle, le principe d'équité professionnelle a été mis en place en 1986 avec la loi de l'équité en matière d'emploi. Cependant, des secteurs tels que le secteur de santé et de l'assistance sociale et le secteur manufacturier sont encore caractérisés par une forte inégalité de répartition des deux genres. Les femmes au Canada sont sous représentées avec moins d'un tiers dans des postes d'échelon supérieur.

L'objectif de ce rapport est de de quantifier l'inégalité du genre dans le milieu professionnel. Un objectif général de notre stage est de voir comment s'implique la linguistique dans les sciences sociales, humaines et technologiques. Par conséquent, la problématique est formulée comme suit : comment s'adresser aux genres en recrutement ? Comment s'adresse les recruteurs aux hommes et aux femmes dans leurs annonces ? Les offres d'emploi sont-elles paritaires?

Pour répondre à notre question de recherche, nous optons pour une approche quantitative et qualitative. Nous analysons un corpus constitué de 232 annonces, 148 offres « secrétaire » et 84 offres « conducteur », publiées sur Indeed Canada en Avril 2022, collectées directement depuis le site Web.

Les résultats de notre recherche montrent que les offres d'emploi pourraient être biaisées suivant le stéréotype du métier sauf dans le cas des postes à échelons élevés où nous remarquons un biais du genre masculin.

Cette étude n'est qu'un point de départ pour une étude ultérieure plus approfondie permettant la création d'un modèle de détection et mesure de biais de genre plus puissant et une étude expérimentale sur l'effet de l'écriture épicène sur les candidats dans les offres d'emploi.

Mots-clés : offre d'emploi, biais du genre, Indeed Canada, équité professionnelle, stéréotype genré.

Scientific abstract

Canada, a country known to be very concerned about professional equity and parity, has implemented the principle of professional equity in 1986. However, there is still a strong inequality in the distribution of both genders in sectors such as health, social assistance, and manufacturing. Women in Canada are underrepresented with only a third of high-level positions being women.

The objective of this report is to quantify gender inequality in the workplace. The aim of the internship is to apply linguistic methods within the fields of social sciences and humanities. Therefore, we formulate our research question as follows: how to evaluate gender representation in the process of recruitment? Are the job offers fair in terms of gender representation?

To answer our research question, we opt for a quantitative and qualitative approach. We analyze a corpus of 232 job offers, 148 "secretary" offer's and 84 "driver" offer's, published on Indeed Canada on April 2022.

The results of our research show that job offers might be biased according to the job stereotype except for high-level positions, for which we observe a male gender bias.

This study is only a starting point for an in-depth study allowing the creation of a more powerful gender bias detection model and an experimental study on epicene writing effect on candidates.

Keywords: job offer, gender bias, Indeed Canada, professional equity, gender stereotype

Résumé « grand public »

Comment s'adressent les recruteurs canadiens aux hommes et aux femmes dans leurs annonces ? Les offres d'emploi sont-elles paritaires? Représentent-elles les femmes et les hommes de façon égalitaire ? Les offre d'emploi des métiers stéréotypés sont-elles biaisées ? Cette étude quantifie l'inégalité du genre dans les offres d'emploi.

Le Canada, malgré sa forte volonté d'établir l'équité professionnelle, se trouve encore avec une division genrée du travail très marquée. Les femmes au Canada sont moins présentes dans des postes à haute responsabilité.

Afin de répondre aux questions précédemment posées, nous analysons des offres d'emploi de secrétaire et de conducteur pour quantifier l'inégalité dans ces deux métiers stéréotypés.

Nos résultats indiquent que les métiers stéréotypés sont biaisés suivant le stéréotype de genre sauf dans le cas d'un grade plus élevé où nous avons remarqué un biais en faveur des hommes.

Mots-clés : offre d'emploi, Canada, stéréotype, inégalité professionnelle.

Lay abstract

How do Canadian recruiters address men and women in their job offers? Are job offers gender equal? Do they represent women and men equally? Are stereotyped professions job offers biased? This study quantifies gender inequality in job vacancies.

Canada, despite its strong desire to establish professional equity, still finds itself with a very marked gendered division of labour. Women in Canada are less present in high responsibility positions.

In order to answer the questions previously asked, we analyze secretaries and drivers job offers to quantify the inequality in these two stereotyped professions.

Our results indicate that stereotyped professions are biases according to their respective gender stereotype, except when the position is high-ranked, in which case we observe a bias in favor of men.

Keywords: job offer, Canada, stereotype, professional inequality.

INTRODUCTION GENERALE

INTRODUCTION GENERALE

La question d'égalité de genre a fait couler beaucoup d'encre et a connu un réel progrès et des critiques depuis des années. Le droit international relatif aux droits de l'homme impose aux états membres l'obligation d'éliminer la discrimination à l'égard des femmes et des hommes dans tous les domaines de leur vie en prenant des mesures pour lutter contre les stéréotypes sexistes dans la vie publique et privée et de s'abstenir de tout stéréotype sexuel[1].

Selon l'encyclopédie canadienne[2] « L'égalité des genres est le principe selon lequel chaque personne, quel que soit son genre, mérite d'avoir les mêmes droits et privilèges ». Si l'égalité des genres consiste à permettre aux personnes peu importe leurs genres de jouir des mêmes occasions et du même soutien, l'équité des genres défend que les personnes de tous les genres doivent bénéficier de différents niveaux de soutien pour être véritablement égales.

Au Canada, un pays connu pour être très soucieux d'équité et de parité professionnelle, 82% des entreprises confirment que la mixité est une priorité et un enjeu important (Mckinsey, 2019). Le principe d'équité professionnelle a été mis en place en 1986 avec la loi de l'équité en matière d'emploi qui vise l'équité professionnelle en vue des quatre groupes : les femmes, les peuples autochtones, les personnes handicapées, et les membres des minorités visibles. Cependant, les femmes sont sous représentés avec moins d'un tiers dans des postes d'échelon supérieur selon les chiffres de la même étude. D'après une autre étude (Appelbaum et Emadi-Mahabadi, 2022) sur les effets de la pandémie sur la parité du genre dans le milieu de travail, la place de la femme est de plus en plus menacée. 5,4 millions d'emploi ont été perdus pour les femmes depuis le début de la pandémie aux Etats-Unis selon le National Women's Law Center (NWLC)[3].

Afin d'apporter plus d'éclairage sur la question, ce rapport se veut une étude quantitative et qualitative des écarts de genre dans les textes des offres d'emploi en utilisant des techniques de fouille de texte aussi dit *Text Mining*. Il est le résultat de nos missions de stage en sein du

[1] Convention sur l'élimination de toutes les formes de discrimination à l'égard des femmes 1981 établit par L'Organisation des Nations Unies.

[2] Andrea, E. (2020). *Égalité des genres.* L'encyclopédie canadienne. [En ligne].

[3] Une organisation à but non lucratif qui défend les droits des femmes et da la communauté LGBTQ.

laboratoire Dynamique Du Langage 'DDL', un établissement public mixte de recherche du CNRS et de l'université Lumière Lyon 2. Notre objectif est de de quantifier l'inégalité de genre dans le milieu professionnel. Un objectif général de notre stage est de voir comment s'implique la linguistique dans les sciences sociales, humaines et technologiques.

Avant d'aller plus loin dans notre étude, nous sommes allés sur le site d'Indeed afin de vérifier si le site a un algorithme qui peut détecter le biais dans le texte de l'annonce. D'après notre expérience, Indeed donne la liberté à l'utilisateur d'écrire son texte sans forcément alerter d'un biais de genre. Cependant, il propose en tapant l'intitulé du poste, un titre avec l'abréviation H/F pour Homme/ Femme.

À la suite de notre expérience, nous avons décidé d'analyser les textes des annonces d'emploi publié sur Indeed Canada. Aujourd'hui, les femmes au Canada constituent 85% des employés de soutien administratif, mais occupent seulement 28% des employés du secteur manufacturier (Mckinsey, 2019). L'attention sera donc mise sur deux métiers : le métier de secrétaire avec une forte présence féminine et le métier de conducteur avec une forte présence masculine. Notre choix de corpus est justifié par les statistiques de Mckinsey déjà mentionnés et l'inégalité de la répartition des hommes et des femmes sur ces deux métiers, représentée par deux tableaux des vingt professions les plus fréquentes chez les femmes et chez les hommes, publiés par l'institut national de la statistique et des études économiques 'INSEE'[4] en 2017. Les deux métiers sont aussi stéréotypés féminin et masculin selon l'étude de Misersky et al. (2014)[5]. Notre corpus est constitué de 232 annonces publiées sur Indeed Canada en Avril 2022, collectées directement depuis le site Web.

Nombreuses et diverses sont les motivations qui nous ont poussé à choisir ce thème. Il s'agit d'un sujet qui a été traité de différentes perspectives. Des études précédemment réalisées ont mis l'accent sur ce sujet, nous citons, comme exemple, l'article de Névéol et al (2022) sur le biais dans les modèles de langue ainsi que l'article de Richy & Burnett (2021) sur les effets des stéréotypes et le genre grammatical dans le biais masculin. D'autres travaux sont en cours

[4] Simon, G. K. (2020). Écarts de rémunération femmes-hommes : surtout l'effet du temps de travail et de l'emploi occupé (publication n° 1803). Insee.

[5] https://link.springer.com/article/10.3758/s13428-013-0409-z/tables/3

dans le cadre du projet GEM (Gender Equality Monitor)[6] qui vise l'analyse automatique des représentations et le traitement des hommes et des femmes dans les médias. Le projet Gendered News[7] nous a été une source d'inspiration pour notre choix de sujet. Il s'agit d'un site web qui mesure quotidiennement les inégalités de mention et de citation des hommes et des femmes dans les médias français. Notre travail se veut une étude du sujet d'un point de vue linguistique.

De ce qui précède, nous formulons la problématique suivante :

- Comment les genres sont-ils représentés en recrutement ? Comment s'adresse les recruteurs aux hommes et aux femmes dans leurs annonces ? Les offres d'emploi sont-elles paritaires?

L'hypothèse qui en découle est la suivante :

- Les offres d'emploi pourraient être biaisés suivant les stéréotypes.

Ainsi pour répondre à notre problématique et confirmer ou infirmer notre hypothèse, nous avons structuré notre travail comme suit :

Premièrement, une introduction générale décrivant le contexte général et particulier de notre sujet ainsi que la problématique, l'hypothèse, la motivation, l'objectif et la structure de notre rapport de stage. Deuxièmement, un chapitre intitulé « Contexte général », où nous allons mettre la lumière dans la première section sur le contexte du stage et l'établissement d'accueil. Dans la deuxième section, nous verrons en détails le projet EVOGRAM qui finance ce stage, les différentes techniques utilisées ainsi que la question de la langue française et féminisation linguistique. Troisièmement, nous allons voir la partie pratique où nous présenterons nos données et décrirons la méthodologie choisie puis nous interpréterons les résultats dans un troisième chapitre.

[6] https://anr.fr/Project-ANR-19-CE38-0012

[7] https://gendered-news.imag.fr/genderednews/

CHAPITRE I

Contexte général

SECTION 1

Contexte de stage

Nous avons réalisé notre stage de fin d'étude de six mois au sein du laboratoire "Dynamique Du Langage" DDL de Lyon. Cet établissement est le fruit d'une collaboration de recherche entre le CNRS et l'Université Lumière Lyon 2 avec une approche interdisciplinaire reposant principalement sur l'étude de la diversité linguistique et les capacités cognitives humaines. Cette première section est consacrée à la présentation de l'établissement d'accueil, à ses différents axes, aux entretiens menés avec ses membres ainsi qu'aux tâches qui nous ont été confiés.

1. L'établissement d'accueil

Le laboratoire Dynamique Du Langage se situe au niveau de l'Institut des Sciences de l'Homme de Lyon. Depuis sa création en 1994, le DDL a élargi ses recherches, partant à la base d'un noyau initial centré sur la phonologie des langues africaines et l'acquisition du discours chez l'enfant en allant vers la linguistique et la psycholinguistique. Aujourd'hui, les langues en danger, les pathologies du langage et la compréhension du développement linguistique de l'enfant sont au cœur des recherches menées par ce laboratoire. Le DDL est rattaché à deux écoles doctorales : 'Lettres, Langues, Linguistique et Arts' et 'Neurosciences et Cognition' et à deux instituts du CNRS : 'l'Institut des sciences humaines et sociales' et 'l'Institut des sciences biologiques' d'où vient sa dimension pluri- et interdisciplinaire.

En recherche, l'entretien est une des méthodes les plus utilisés pour différentes études. D'après Romelaer (2005) « L'entretien est une des méthodes qualitatives les plus utilisées dans les recherches en gestion. Un entretien de recherche n'a rien de commun avec une discussion dans laquelle on se laisse porter par l'inspiration du moment. ». Un entretien de recherche permet une collecte de données et d'informations pour les analyser ensuite. En fonction de notre recherche se définit le type de l'entretien le mieux adapté. Nous avons mené, dans le cadre de notre stage, trois entretiens exploratoires avec différents membres du DDL afin de découvrir de plus près l'activité du laboratoire.

2. Les axes du laboratoire

Le laboratoire DDL se caractérise non seulement par sa diversité linguistique mais aussi par sa diversité humaine. Il regroupe des enseignants chercheurs, des chercheurs, des doctorants et des stagiaires de différents pays. Il est organisé autour de trois axes de recherche DENDY, DILIS ainsi que COVALI regroupant les principales thématiques qui le structurent.

2.1. DENDY 'Développement, Neurocognition, Dysfonctionnements' :

Comme son nom l'indique, cet axe a pour objectif principal d'étudier le développement et le traitement du langage chez des individus de différents âges avec ou sans troubles pathologiques (dyslexie, trouble développemental du langage oral, maladie d'Alzheimer, parleurs tardifs etc..). Il adopte une approche pluridisciplinaire et pluri-méthodologique. Deux principales thématiques sont étudiées au sein de cet axe : d'une part, les liens entre le langage et motricité. D'autres part, la variation entre les locuteurs et leurs différentes pratiques langagières dans un contexte sociolinguistique.

Plusieurs outils sont utilisés pour recueillir et traiter les données langagières au sein de DENDY, nous citons :

- L'enregistrement de corpus audio-vidéo à l'aide d'un enregistreur et d'une caméra qui permettent un recueil de données spontanés (sans élicitation).
- L'électroencéphalographie (EEG) qui est un examen qui servant à enregistrer l'activité du cerveau par le biais d'électrodes placées sur le cuir chevelu.
- L'axe a aussi développé ses propres outils comme les inventaires français du développement communicatif 'IFDC' qui sont des questionnaires destinés aux parents afin de tracer le développement langagiers et gestuels chez les enfants.

Interview :

Nous avons questionné Lucie Métral, Doctorante et membre de l'équipe DENDY pour avoir une idée sur ses recherches menées au sein du laboratoire. Lucie a commencé sa thèse de doctorat en octobre 2021 après avoir fait une Licence et un Master en Sciences du langage à l'Université Grenoble Alpes (UGA).

Son sujet de Thèse s'intitule "Évaluation de l'effet du Baby Sign sur la communication adulte-enfant et sur le développement langagier en crèche", dans lequel, elle souhaite déterminer si le Baby Sign a un impact sur le développement langagier et la communication chez l'enfant vu que cet outil n'a pas encore relevé d'effets prégnants dans la littérature.

Pour tester son hypothèse, elle a choisi une méthode expérimentale qui dure un an, pendant cette période, elle va faire des évaluations de langage sur deux groupes de crèches : un groupe témoin et un groupe expérimental pour avoir des données comparables. Chaque groupe comprend quatre crèches et chaque crèche est visitée tous les trois mois pendant un an.

L'expérimentation passe par trois étapes :

- La pré-expérimentation : consiste à distribuer des questionnaires sociodémographiques aux parents et aux professionnels ainsi qu'à filmer des vidéos d'observation en crèche.
- L'expérimentation : se résume en deux jours de formation des professionnels au Baby-Sign par l'association « Signe Avec Moi » et le pratiquer avec les enfants, éventuellement le transmettre aux parents s'ils le souhaitent.
- La post-expérimentation : réside en la collecte de données en filmant les interactions deux semaines après puis trois, six, neuf et douze mois. A six et à douze mois, Lucie redistribue des questionnaires langagiers aux parents pour connaitre l'avancée de l'enfant. A douze mois elle questionne les professionnels en entretien semi-dirigé et leur distribue des questionnaires pour savoir leurs pratiques et leurs ressentis par rapport à l'outil.

Afin de récolter toute donnée sur l'enfant, le consentement des parents est obligatoire. Les données vidéos seront transcrites par la suite (mots, gestes, regard, posture, signes...) sur le logiciel Elan. Un recours au logiciel R est possible pour faire des statistiques.

2.2. DILIS 'Diversité Linguistique et ses Sources' :

L'axe DILIS a vu le jour à la suite de la fusion des deux anciens axes DTT 'Description, Typologie, Terrain' et HELAN2 'Histoire et Ecologie du Langage et des Langues'. Les principales contributions de cet axe concernent la description des langues peu ou pas décrites et la typologie des langues, les langues en danger et l'origine de la diversité linguistique. DILIS propose régulièrement un séminaire composé de quatre ateliers traitant les thématiques de l'axe afin de discuter des travaux de recherches en cours.

Interview :

Nous avons interviewé Matthew Stave, post doctorant participant à DILIS. Matthew est titulaire d'un B.A. double mention Littérature et Langue espagnole de l'Université George Fox, États-Unis obtenu en 2001 et un doctorat en Linguistique de l'Université d'Oregon, États-Unis. L'intitulé de sa thèse était 'les relations temporelles de la narration verbale et non verbale dans la narration : étude gestuelle et multimodale codée (parole, comportement, geste manuel et de la tête pour les locuteurs et les auditeurs sur ELAN).

Matthew travaille sur le projet DoReCo language 'Documentation Reference Corpora'[8], un projet collaboratif franco-allemand démarré en 2019 qui consiste à rassembler un corpus de cinquante langues diversifiées souvent en voie de disparition déjà collectées et transcrites sur le logiciel Elan. Le travail de Matthew consiste à effectuer l'alignement du temps phonémique à l'aide de Maus[9], un site web d'étiquetage et de segmentation phonétique automatique, puis mapper le temps des phonèmes aux morphèmes. Le fichier Elan final comporte la transcription et la traduction, les morphèmes, les parties de discours et la durée de chaque phonème. Avec François De Lafontaine, expert en Python, il a développé un package Python afin de lire et manipuler les données et exporter le fichier Elan sous différents formats. Le projet, qui sera publié en juillet 2022, a pour objectif d'apporter une certaine sensibilisation aux langues moins comprises et mettre à disposition de la recherche un corpus linguistique (phonologiques et morphologiques) professionnel. Ce corpus permettra de répondre à des questions comme combien de morphèmes y a-t-il dans une langue ? Combien y a-t-il de fonctions grammaticales dans le morphème ?

2.3. COVALI 'COntraintes perceptivo-motrices et VAriation LInguistiques' :

Cet axe est considéré comme un thème transversal exploratoire unissant les compétences des deux axes DENDY et DILIS déjà cités. Les membres de COVALI veulent réunir leurs connaissances et compétences en matière de typologie sémantique de l'expression de la trajectoire et dans le domaine de la cognition sensori-motrice. L'axe comprend que cinq participants mais aucun doctorant qui travaille sur un sujet à temps ce qui fait que les recherches avancent doucement.

Interview :

Nous avons effectué un bref entretien avec Alice Roy, chargée de recherche première classe et responsable de l'axe COVALI pour mieux comprendre les recherches effectuées au sein de cet axe. D'après Alice, quatre projets exploratoires sont menés à COVALI. L'un d'eux est le projet « projectoire » né à partir des observations de régularités en typologie sémantique dans des langues de différentes familles qui partagent la description de la trajectoire et le déplacement sur un axe vertical.

[8] http://doreco.info/project/

[9] https://www.phonetik.uni-muenchen.de/forschung/Verbmobil/VM14.7eng.html

3. Les tâches effectuées

Au cours de nos six mois de stage au DDL, nous avons eu l'opportunité de découvrir et de s'initier à la fouille et à l'analyse de texte en travaillant sur le projet EVOGRAM.

Les missions qui nous ont été confiées consistaient d'abord à collecter et créer un corpus de données. Puis, entraîner et tester différents modèles de classification pour détecter des informations ciblées à partir de la base données. Ensuite évaluer les modèles sur différentes tâches de classification. Nous avons fait recours au langage de programmation R et Python afin de traiter nos données. Nous nous sommes initié aux deux logiciels avant le début de stage. Des réunions régulières tous les quinze jours avec nos tuteurs de stage ont été mis en place afin de suivre l'avancée de notre travail. Lors de la réunion, nous discutons des tâches déjà effectuées et des tâches à venir. Nous avons commencé par l'apprentissage des différentes bibliothèques R nécessaires pour la collecte. Ensuite, nous avons abordé le traitement et l'analyse de nos données. Puis, nous avons effectué le développement du code.

En somme, notre séjour au laboratoire Dynamique Du Langage fût une des expériences les plus enrichissantes en terme d'apprentissage. Le DDL a la spécificité d'être une mosaïque rassemblant des chercheurs des quatre coins du monde travaillant sur différentes langues minoritaires ainsi que d'autres thématiques. Grace à ce stage, nous avons pu discuter régulièrement avec des chercheurs de différents champs de la linguistique et découvrir le TAL (Traitement Automatique des Langues), un domaine qui nous intéresse et qui permet l'application de nos connaissances en matière de science de langage en faveur d'autres disciplines.

SECTION 2

Cadrage théorique

Notre stage fut une découverte d'un autre champ de la linguistique : le traitement automatique du langage abrégé TAL. Tout au long de ce stage, nous avons découvert et appliquer des nouvelles techniques pour la première fois. Cette deuxième section se veut une introduction au projet EVOGRAM sur lequel nous travaillons et un cadrage théorique des différentes techniques utilisé pendant ce stage ainsi que la notion du genre en langue française.

1. Projet EVOGRAM

EVOGRAM[10] est un projet de recherche d'une durée de deux ans, financé par l'ANR en partenariat avec le laboratoire Dynamique Du Langage et coordonné par Marc ALLASSONNIÈRE-TANG.

Intitulé « L'effet des facteurs linguistiques et non-linguistiques sur l'évolution des systèmes de classification nominale » EVOGRAM se veut une analyse quantitative de l'influence des facteurs linguistiques : (morphologie, syntaxe, phonologie, sémantique) et non linguistiques : (biais cognitive, taille et structure de la population, l'environnement naturel) sur l'émergence et l'évolution des systèmes de classification nominale qui comprend les genres grammaticaux, les classes nominales et les classificateurs.

Pour ce stage, nous avons décidé de nous concentrer sur les facteurs non linguistiques, plus précisément le biais cognitif, pour étudier le biais du genre dans les offres d'emploi et mesurer quantitativement et qualitativement l'inégalité de genre dans le milieu professionnel.

2. Techniques de traitement de données

Pour notre stage, nous avons utilisé différentes techniques que nous allons aborder maintenant.

2.1. Data Scraping

Le Grattage de données ou Data Scraping est une technique d'extraction et collecte de données de sources non ou mal structurés à partir d'un format lisible par l'homme (*human-readable format)* d'un autre programme informatique. Cette technique utilise des langages de programmation ou des outils d'extraction de données.

[10] https://anr.fr/Projet-ANR-20-CE27-0021

Trois étapes constituent le processus : premièrement, la sélection de la source de données ciblées. Par exemple, un site Web. Ensuite, la collecte des données, qui dépend de l'outil et du type de technique employée. Finalement, la conservation et le stockage des données. Par exemple, quel format utiliser ? Excel, CSV, HTML ou autre.

Il existe trois variantes de technique de *Data Scraping* : Le *Screen Scraping,* le *Report Mining* et le *Web Scraping* qui nous intéresse ici et que nous allons détailler par la suite.

Le *Screen Scraping* est un grattage d'écran pour capturer ou copier des informations s'affichant sur un écran (poste de travail, application etc), automatiquement à l'aide de scripts ou de logiciels de scrap. Cette technique très utilisée dans le domaine bancaire permet de convertir les données d'une ancienne application à une nouvelle.

Le *Report Mining* consiste à gratter des données non pas à partir d'un écran mais des rapports informatiques tels que des fichiers en format lisible par l'homme comme les textes ou les PDF. C'est une approche simple et rapide pour obtenir des données sans passer par le système source comme le nécessite le *Screen Scraping.* Nous avons déjà testé cette technique sur un fichier PDF. Nous avons fait recours à deux script R et Python afin d'extraire et compter des occurrences de mots-clés dans un texte de grammaire linguistique. Pour cela nous avons utilisé la librairie pdf plumber de Python et la librairie pdftools de R.

2.1.1. Web Scraping

Très similaire au *Screen Scraping* qui collecte des données affichées sur l'écran, le web scraping est une technique d'extraction d'informations qui gratte des données publiées seulement sur le web. Donc, à partir de pages Web construites à l'aide de langages de balisage textuels (HTML et XHTML).

Le *Web Scraping* peut être automatique en utilisant un bot ou un robot d'exploration Web ou « grattoir Web ». Il peut être utilisé pour des non-programmeurs en copiant et collant les données d'une page Web dans un fichier. L'examen manuel et le copier-coller d'un humain peut être meilleur que le grattage automatique si les données à gratter n'ont pas de volume important ou si les sites mettent explicitement des barrières empêchant l'extraction automatique.

Le grattage de données peut être contraire aux conditions générales d'utilisation. Les sites ou les entreprises n'ont pas intérêt à permettre l'accès à leurs contenus pour être utilisés à

des fins inconnues ou non autorisées, ils ne mettent pas alors tous leurs données sur l'API (Application Programming Interface). Pour entraver ou contrecarrer le processus de scraping, ils sécurisent le site à l'aide de systèmes de protection comme le captcha qui est difficilement surmonté par un navigateur ou en modifiant régulièrement leur balisage HTML ou même en incorporant le contenu en objets media ce qui nécessite une reconnaissance optique de caractères (OCR).

Dans la tentative de récupérer des données, les scrapeurs risquent de tomber sur ces systèmes de protection. Un code qui veut extraire trente pages par seconde tombera surement sur un ou plusieurs captchas.

Notre premier plan était d'étudier le cas de l'inégalité de genre en France. Mais, lors de l'application de la technique *Web Scraping* et en s'entrainant au code, Indeed a activé des Captchas sur son site. Nous avons donc fait face à ce même problème avant de décider de passer à Indeed Canada.

Pour un grattage de données réussi il faut être prudent. Pour tromper les systèmes de sécurité mis sur le site nous pouvons insérer un temps de pause dans notre code grâce à la fonction sleep(). Ceci permet de simuler un comportement humain comme les clics ou passer par différentes adresses IP grâce au module Key de python. Une autre alternative serait de scraper avec Selenium qui est à la base un testeur de page web. Il crée des robots pour simuler une activité normale d'un vrai utilisateur qui navigue dans des pages web pour éviter l'extraction rapide et donc les captchas. Pour ce faire, il faut utiliser la Librairie RSelenium dans R et module selenium de Python.

Dans notre cas, nous avons effectué un scrap prudent après avoir choisi Indeed Canada. Nous avons testé notre code bout par bout pour s'assurer qu'il extrait les bonnes informations avant de lancer le programme pour nous éviter de tourner le code plusieurs fois et que le scrap soit détecté par le site.

2.2. Text Mining

La fouille de texte, aussi appelée *Text Mining* en anglais, est un ensemble de techniques qui permettent l'exploration de données non structurés (texte) afin de récupérer des informations utiles exploitables et éventuellement d'identifier des régularités et des tendances dans le texte en langage naturel. Elle utilise des approches de linguistique, de statistique et

d'informatique. Elle repose sur le *Machine Learning*[11] pour repérer des modèles ou des tendances et en tirer des valeurs.

Le *Text Mining* est très utilisé par différents secteurs de recherche ou en marketing par des entreprises pour étudier le comportement des consommateurs par exemple. Il repose donc sur des algorithmes qui peuvent balayer les textes afin des retrouver des modèles ou des relations qui permettent la classification des documents, l'analyse des sentiments ou d'intentions.

Une des techniques de fouille de texte est l'extraction d'information. Elle consiste à extraire automatiquement des informations, à l'aide de la reconnaissance d'entités nommées (*Named-entity recognition)*. Cette méthode consiste à qui elle chercher des objets textuels dans des catégories tels que les noms, les lieux, les quantités, les dates etc. l'extraction d'information peut être réussite si les entités recherchées sont spécifiques à une catégorie sinon une désambiguïsation d'entité s'impose.

Pour notre travail, et à cause de la contrainte de temps, nous avons eu recours aux expressions régulières dit *Regex* afin d'extraire les différentes informations ciblées.

2.3. Data Mining

L'exploration ou le fourrage de données « *Data Mining* » est l'analyse de données de différentes perspectives. Il comprend plusieurs techniques, nous avons utilisé les suivantes :

2.3.1. La visualisation des données

C'est le processus d'interprétation visuelle rapide et efficace des données à l'aide des outils graphiques simples tels que les histogrammes ou les diagrammes pour mieux comprendre les résultats. L'objectif est de faciliter l'identification des modèles, des tendances et des valeurs aberrantes dans les données. C'est une étape qui vient une fois les données sont collectées, traitées et modélisées, pour ensuite être visualisées pour en tirer des conclusions.

Au début de la visualisation, le logiciel le plus utilisé pour la visualisation était Microsoft Excel pour transformer les informations en un tableau, un graphique à barres ou un graphique à secteurs. Bien que Excel soit encore populaire, d'autres logiciels sont désormais disponibles,

[11] Dit apprentissage automatique en français est une sous-catégorie de l'intelligence artificiel IA. Il permet aux machines d'« apprendre » à partir des données pour améliorer leurs performances.

notamment le langage de programmation R qui est conçu pour les mesures statistiques. C'est ce dernier que nous avons utilisé.

2.3.2. La corrélation

La corrélation est une méthode statistique utilisée pour évaluer une éventuelle dépendance ou relation linéaire entre deux variables continues. Elle est simple à calculer et à interpréter. Le coefficient de corrélation qui est la force de l'association linéaire des deux variables varie entre -1, ce qui signifie une corrélation négative parfaite, à 0 pas de corrélation et +1 une corrélation positive parfaite. En résumé, plus la corrélation est forte, plus le coefficient de corrélation se rapproche de +1, dans ce cas-là lorsque la valeur d'une variable augmente, la valeur de l'autre a également tendance à le faire. Si, d'autre part, le coefficient est un nombre négatif, les variables sont inversement liées, ce qui veut dire lorsque la valeur d'une variable augmente, la valeur de l'autre a tendance à baisser. Dans le cas d'un coefficient de corrélation 0 ceci indique qu'il n'existe aucune relation linéaire entre deux variables continues. Le langage de programmation R, nous permet de visualiser la corrélation à l'aide d'un corrélogramme en utilisant le package corrplot ou GGally.

2.3.3. La régression

Très utilisée en statistique, la régression linéaire est un des modèles de régression le plus connu. Il sert à établir une relation entre deux variables : une variable dépendante et l'autre indépendante. La régression est utilisée pour estimer la valeur d'une variable sur la base d'une autre variable. Pour effectuer une régression, nous pouvons avoir recours à Microsoft Excel ou un des langages de programmation tels que R ou Python.

La régression et à la corrélation sont deux notions souvent confondues or qu'elles sont différentes. La corrélation est la mesure de relation entre deux variables donc l'intensité de cette relation, elle donne une valeur numérique qui exprime la force de cette relation. Cependant, la régression détermine le mode d'association entre ses deux variables donc la relation d'une variable par rapport à l'autre.

2.3.4. Les arbres décisionnels

L'arbre de décision est un outil de classification et de décision sous forme graphique d'arbre représentant les choix sur les branches et les décisions sur les feuilles. Il sert à prédire une valeur ou une catégorie. Cette méthode appartient au *Machine Learning* et elle est très

populaire en *Data Science*. L'arbre de décision utilise un algorithme dit supervisé qui va construire un arbre à partir des données numériques ou catégorielles. L'arbre de décision est intuitif, facile à comprendre et interpréter. Il nécessite très peu de préparation de données comme la normalisation. Cependant, si l'arbre devient très complexe, il peut générer un sur apprentissage ou un biais si une ou certaines classes sont dominantes.

Pour évaluer la performance d'un modèle de classification, nous pouvons faire différents calculs afin d'avoir un taux de réussite ou de prédiction, de précision ou de rappel.

3. Le français canadien et le français de France

La différence entre le français de France et le français du Canada est assez évidente au niveau oral vu l'accent tonique et le vocabulaire assez variés sauf qu'au niveau écrit ces évidences deviennent moins transparentes. La position des francophones canadiens anti-anglicistes a fait naitre un vocabulaire de traduction littérale depuis l'anglais qu'on ne trouve pas en français de France tels que « pourriel » pour dire « *spam* » ou « infonuagique » pour « *cloud computing* ». On peut aussi trouver des mots propres au français canadien, d'autres à signification différente ou à plusieurs significations, l'exemple de « pastèque » en France appelé « Melon d'eau » au Canada. Cependant, au niveau syntaxique et de la grammaire, les deux variétés de la langue de Molière restent identiques.

4. La langue française et la féminisation linguistique

Suite aux changement sociétaux et aux mouvement féministe, la question de la féminisation linguistique en emploi en langue française s'est posée sur table et a été traité de différentes manières selon le pays.

En France, le guide « Femme, j'écris ton nom… : guide d'aide à la féminisation des noms de métiers, titres, grades et fonctions » (Becquer et al.) est apparu en 1999 après la réactivation du circulaire publié au Journal officiel par l'Assemblée nationale en 1998 qui vise à combler les lacunes du vocabulaire en matière de noms de métiers, de titres et de fonctions au féminin. Selon ce guide, l'usage du masculin générique en toute situation est une perte de la richesse de la langue ainsi que son usage générique pour une situation particulière et singulière comme pour dire « madame le ministre » est considéré comme une agression aux femmes et opposé à la logique, la grammaire et la civilité (pp. 39).

L'Académie française[12], cependant, a joué un rôle moralement bloquant envers ce changement linguistique en le qualifiant de « péril mortel ». Ce n'est qu'en 2019 que l'académie valide la féminisation des noms de métiers et de fonctions.

Le Canada, un pays soucieux de l'équité professionnelle, s'est préoccupé par la question du genre et la féminisation linguistique à l'arrivé du mouvement féministe nord-américain dans les années 1970. Selon Arbour et al (2014), un réaménagement linguistique de ses deux langues officielles se met en place suite au même mouvement linguistique en Amérique du Nord. Ce réaménagement vise à mieux représenter les femmes en emploi. Sauf qu'au niveau morphologique, la langue française et anglaise ne suit pas les mêmes règles. C'est là où l'Office Québécois de la Langue Française (OQLF)[13] intervient pour jouer un rôle primordial en la question de féminisation des appellations d'emploi. Une première proposition était la « Gazette officielle du Québec » en 1979 qui sert comme avis linguistique officiel aux administrations pour recommander l'emploi des variantes féminines et l'accord du déterminant au féminin.

En 1981 l'office propose le recours aux termes génériques et aux tournures neutres, et l'utilisation d'appellations en toutes lettres comme principes de base de la féminisation des textes comme solution syntaxique. Une autre proposition apparait en en 1986 intitulée « Titres et fonctions au féminin : essai d'orientation de l'usage » suivi par un guide aux québécois qui s'intitule « Au féminin : guide de féminisation des titres de fonction et des textes nouvelles pratiques d'écriture » publié en 1991.

4.1. Ecriture épicène ou rédaction inclusive ?

Selon l'Office québécois de la langue française, l'écriture épicène «consiste à éviter les genres grammaticaux masculins et féminins en ce qui concerne les personnes, sans toutefois faire appel à des néologismes, au contraire de la rédaction non binaire» (Office québécois de la langue française, 2018). C'est donc une rédaction qui permet une représentation équitable des femmes et des hommes en utilisant différents procédés et tout en

[12] C'est une institution à autorité morale fondée en 1634 qui a pour fonction la veille sur la langue française et son perfectionnement.

[13] C'est une institution publique, anciennement appelée Office de la Langue Française (OLF) qui a pour fonction l'officialisation linguistique, les recommandations terminologiques et de la francisation de la langue de travail des secteurs public et privé.

préservant la lisibilité et l'intelligibilité du texte. Le résultat : un texte qui respecte la présence des deux genres, lisible et compréhensible.

4.2. Les règles de l'écriture épicène

L'office québécois de la langue française a publié ses guides de rédaction épicène « Avoir bon genre à l'écrit : guide de rédaction épicène » (Vachon-L'Heureux et Guénette 2006) et « Le français au bureau » (Guilloton et al. 2014) qui proposent des règles relatives à la formalisation neutre et la féminisation syntaxique des textes et un répertoire d'appellations de personnes au masculin et au féminin. Parmi ces règles :

a. L'abandon de l'emploi systématique du masculin générique qui ne présente pas les personnes et leur genre ainsi que l'abandon de la note explicative qui dit que l'emploi du masculin générique pour objectif d'alléger le texte n'est pas accepté.
b. L'évitement des formes tronquées dit aussi doublets abrégés qui peut nuire à la bonne lecture du texte sauf dans le cas d'un espace restreint où les formes tronquées deviennent une nécessité. Pour écrire en doublets abrégés, l'OQLF recommande l'usage des parenthèses et de point. Cependant d'autres formes sont répandues :
    - Le point : candidat.e
    - Le trait d'union : candidat-e
    - Les parenthèses : candidat(e)
c. L'équilibration du texte en utilisant les différents procédés de féminisation afin de mieux représenter les femmes et les hommes et obtenir un texte compréhensible comme le recours au doublet qui est l'ensemble de la forme féminine et masculine que ce soit dans :
    - Le nom : le candidat et la candidate.
    - Le déterminant : la ou le candidat / il ou elle.

Cependant, il existe des pratiques relatives à l'emploi de doublet :

- L'ordre de doublet reste libre ; La forme masculine peut précéder la forme féminine sauf dans le cas où le doublet est suivi d'un adjectif, dans ce cas-là il est préférable que la forme féminine précédé la forme masculine pour éviter une phrase du type : le candidat ou la candidate sélectionné.
- Dans le cas d'un nom ou un adjectif épicène, la seule répétition du déterminant suffit comme dans : le ou la secrétaire.

- La répétition de l'adjectif ou le participe passé après le doublet n'est pas recommandée, il suffit de l'accorder au masculin. Exemple : la candidate ou le candidat étranger.

d. La modération de l'usage des formules neutres et le recours aux différents procédés pour éviter la dépersonnalisation de texte en utilisant :
   - Les mots épicènes : personne, cadre.
   - Les adjectifs épicènes : apte.
   - Les noms collectifs : personnel, public.
   - Les Pronoms épicènes : vous, plusieurs.
   - Les phrases épicènes : avez-vous un la nationalité canadienne ? au lieu de : êtes-vous canadien ou canadienne ?

CHAPITRE II

Présentation et analyse des données

1. Présentation des données

Nous avons choisi d'analyser un corpus composé de 232 offres d'emploi. 148 offres « secrétaire » et 84 offres « conducteur » publiées sur le site web Indeed Canada en Avril 2022. L'inégalité de la répartition des femmes et des hommes dans les deux métiers selon la déclaration annuelle de données sociales (DADS) et déclarations sociales nominatives (DSN) Insee 2017, ainsi que la représentation mentale que déclenche chacun des métiers selon l'étude de Misersky et al (2014) étaient des critères de choix de notre corpus. Les annonces ont été récoltées en utilisant la technique du *Web Scraping* qui permet l'extraction des informations ou de données à partir du fichier HTML du site. Nous devons signaler que nous avons rencontré des difficultés à gratter les données d'Indeed France, le site a détecté notre opération et a mis en place deux captchas[14], ce qui a causé le disfonctionnement de notre code d'où vient notre choix de basculer vers Indeed Canada. Une alternative aurait été de scraper avec Selenium, faute de temps nous ne l'avons pas fait. Il est important de mentionner que les résultats de recherche des offres peuvent afficher d'autres noms de métiers proche du métier recherché. Par exemple si on cherche « secrétaire » souvent on trouve des offres d'« agent ou d'adjoint administratif ».

Notre choix du site Indeed était évident vu que c'est le site le plus connu en matière de recherche d'emploi. Il se dit le site numéro 1 d'offre d'emploi dans le monde entier avec plus de 200 millions visiteurs uniques par mois. D'après notre expérience, Indeed ne possède aucun algorithme qui détermine si le texte de l'annonce est biaisé ou pas, nous avons donc été sûr que le contenu des offres est naturel et n'a pas été corrigé par le site.

2. Méthodologie choisie

Etant donné que nous traitons un contenu représentationnel, nous avons donc décidé, pour ce présent travail, de suivre une approche quantitative et qualitative. Selon Robert et Bouillaguet (2007) : « L'analyse de contenu se définit comme une technique permettant l'examen méthodique, systématique, objectif et, à l'occasion, quantitatif du contenu de certains textes en vue d'en classer et d'en interpréter les éléments constitutifs, qui ne sont pas totalement accessibles à la lecture naïve ». Le contenu analysé des annonces est dit « naturel » (Dany,

[14] Un test ou système de contrôle d'accès au site.

2016), c'est donc un contenu qui existe déjà sans aucune intervention de notre part. Notre méthode consiste à suivre différentes étapes :

Premièrement, la création d'une base de données de notre corpus par la collecte de contenu en utilisant la technique du Web Scraping. Deuxièmement, l'observation du contenu et le repérage des différents indicateurs de genre dans les textes d'offres d'emploi. Troisièmement, la création de modèles de classification qui permettent d'analyser le corpus et d'extraire les informations demandées. Quatrièmement, le test de la fiabilité ou la validité des résultats de l'analyse quantitative en la comparant avec les résultats l'analyse qualitative et finalement la réalisation de différentes analyses statistiques pour interpréter nos données et les prédire.

2.1. Analyse qualitative

L'analyse qualitative est une méthode très utilisée pour l'analyse de contenu traitant les représentations sociales. Elle consiste, dans notre cas, à dépouiller manuellement tout le corpus, identifier la présence du genre et attribuer un indice de biais à chaque offre. Ceci permet de tester l'exactitude des résultats communiqués par notre modèle.

Pendant l'analyse manuelle de notre corpus, nous avons remarqué que pour exprimer la parité du genre dans les offres d'emploi canadiennes, les recruteurs font recours à plusieurs pratiques:

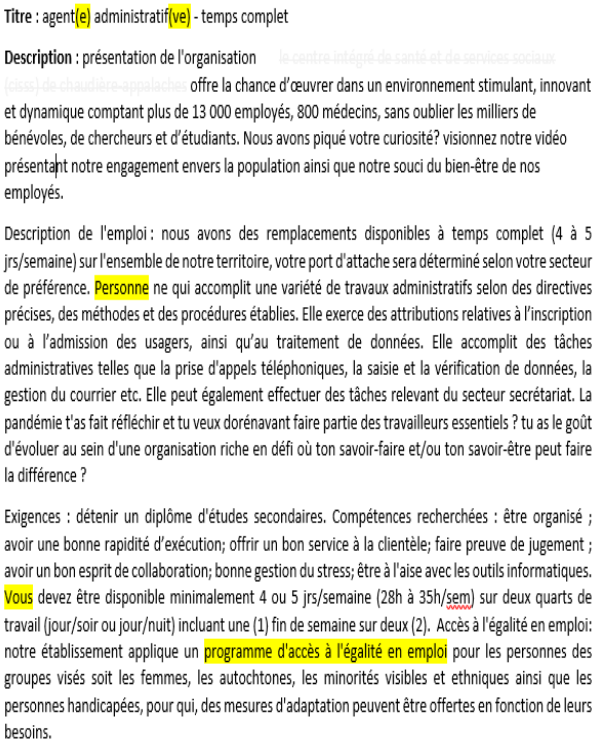
- Le doublet dans les noms de métiers, les adjectifs, les articles définis et indéfinis et le troisième pronom personnel du singulier. Comme dans : Il ou elle, le ou la, un / une, agent / agente, administratif ou administrative.
- Les formes tronquées tels que : (ve), (e), (trice), (euse), (ère) et (ne).
- L'abréviation H/F pour Homme / Femme.
- La précision que l'entreprise s'inscrit au programme de l'équité d'emploi qui comprend quatre groupes dont le groupe des femmes.
- Le recours à la neutralité, les mots et les phrases épicènes.

Toutefois, le biais du genre peut s'exprimer par :

- L'absence de l'abréviation H/F, des formes tronquées, des doublets et des formes épicènes.
- La présence des noms de métier et des désignations adjectivales (nom de métier + adjectif) genrées.

- Note de « décharge de responsabilité » : l'emploi du masculin générique et l'ajout d'une phrase à la fin du texte indiquant que l'emploi du masculin est pour alléger le texte.

Regardons un exemple illustratif d'une offre d'emploi biaisé et une autre paritaire (fig. 1 & 2):



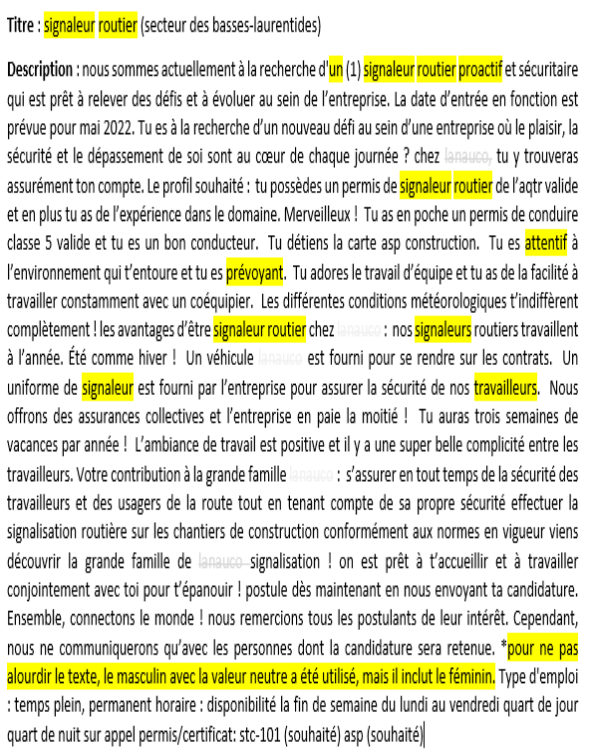
Titre : agent(e) administratif(ve) - temps complet

Description : présentation de l'organisation le centre intégré de santé et de services sociaux (cisss) de chaudière-appalaches offre la chance d'œuvrer dans un environnement stimulant, innovant et dynamique comptant plus de 13 000 employés, 800 médecins, sans oublier les milliers de bénévoles, de chercheurs et d'étudiants. Nous avons piqué votre curiosité? visionnez notre vidéo présentant notre engagement envers la population ainsi que notre souci du bien-être de nos employés.

Description de l'emploi : nous avons des remplacements disponibles à temps complet (4 à 5 jrs/semaine) sur l'ensemble de notre territoire, votre port d'attache sera déterminé selon votre secteur de préférence. Personne ne qui accomplit une variété de travaux administratifs selon des directives précises, des méthodes et des procédures établies. Elle exerce des attributions relatives à l'inscription ou à l'admission des usagers, ainsi qu'au traitement de données. Elle accomplit des tâches administratives telles que la prise d'appels téléphoniques, la saisie et la vérification de données, la gestion du courrier etc. Elle peut également effectuer des tâches relevant du secteur secrétariat. La pandémie t'as fait réfléchir et tu veux dorénavant faire partie des travailleurs essentiels ? tu as le goût d'évoluer au sein d'une organisation riche en défi où ton savoir-faire et/ou ton savoir-être peut faire la différence ?

Exigences : détenir un diplôme d'études secondaires. Compétences recherchées : être organisé ; avoir une bonne rapidité d'exécution; offrir un bon service à la clientèle; faire preuve de jugement ; avoir un bon esprit de collaboration; bonne gestion du stress; être à l'aise avec les outils informatiques. Vous devez être disponible minimalement 4 ou 5 jrs/semaine (28h à 35h/sem) sur deux quarts de travail (jour/soir ou jour/nuit) incluant une (1) fin de semaine sur deux (2). Accès à l'égalité en emploi: notre établissement applique un programme d'accès à l'égalité en emploi pour les personnes des groupes visés soit les femmes, les autochtones, les minorités visibles et ethniques ainsi que les personnes handicapées, pour qui, des mesures d'adaptation peuvent être offertes en fonction de leurs besoins.

Fig. 1 : offre d'emploi paritaire



Titre : signaleur routier (secteur des basses-laurentides)

Description : nous sommes actuellement à la recherche d'un (1) signaleur routier proactif et sécuritaire qui est prêt à relever des défis et à évoluer au sein de l'entreprise. La date d'entrée en fonction est prévue pour mai 2022. Tu es à la recherche d'un nouveau défi au sein d'une entreprise où le plaisir, la sécurité et le dépassement de soi sont au cœur de chaque journée ? chez lanauco tu y trouveras assurément ton compte. Le profil souhaité : tu possèdes un permis de signaleur routier de l'aqtr valide et en plus tu as de l'expérience dans le domaine. Merveilleux ! Tu as en poche un permis de conduire classe 5 valide et tu es un bon conducteur. Tu détiens la carte asp construction. Tu es attentif à l'environnement qui t'entoure et tu es prévoyant. Tu adores le travail d'équipe et tu as de la facilité à travailler constamment avec un coéquipier. Les différentes conditions météorologiques t'indiffèrent complètement ! les avantages d'être signaleur routier chez lanauco : nos signaleurs routiers travaillent à l'année. Été comme hiver ! Un véhicule lanauco est fourni pour se rendre sur les contrats. Un uniforme de signaleur est fourni par l'entreprise pour assurer la sécurité de nos travailleurs. Nous offrons des assurances collectives et l'entreprise en paie la moitié ! Tu auras trois semaines de vacances par année ! L'ambiance de travail est positive et il y a une super belle complicité entre les travailleurs. Votre contribution à la grande famille lanauco : s'assurer en tout temps de la sécurité des travailleurs et des usagers de la route tout en tenant compte de sa propre sécurité effectuer la signalisation routière sur les chantiers de construction conformément aux normes en vigueur viens découvrir la grande famille de lanauco signalisation ! on est prêt à t'accueillir et à travailler conjointement avec toi pour t'épanouir ! postule dès maintenant en nous envoyant ta candidature. Ensemble, connectons le monde ! nous remercions tous les postulants de leur intérêt. Cependant, nous ne communiquerons qu'avec les personnes dont la candidature sera retenue. *pour ne pas alourdir le texte, le masculin avec la valeur neutre a été utilisé, mais il inclut le féminin. Type d'emploi : temps plein, permanent horaire : disponibilité la fin de semaine du lundi au vendredi quart de jour quart de nuit sur appel permis/certificat: stc-101 (souhaité) asp (souhaité)

Fig. 2 : offre d'emploi biaisée

Vu qu'il existe plusieurs facteurs qui décident la parité ou l'inégalité du genre des annonces d'emploi au Canada, nous avons décidé de les prendre tous en considération. Pour notre étude qualitative nous avons donné un score qui varie entre 1, 0 et -1 au titre et à la description. Pour avoir la moyenne de parité ou d'inégalité de genre, nous avons compté la moyenne des deux colonnes titre et description pour avoir une valeur de biais de :

- 1 : biaisé femme.
- 0.5 : moyennement biaisé femme.
- 0 : paritaire.
- -0.5 : moyennement biaisé homme.
- -1 : biaisé homme.

Pour s'assurer de la bonne annotation de notre corpus et vu la non possibilité d'une annotation tierce, nous avons réannoter sur un nouveau fichier 40 offres, 20 de chaque métier, choisi aléatoirement deux mois après la première annotation pour les comparer ensuite. L'extraction automatique nous a aussi aidé à corriger quelques informations que nous avons manqué mais que la machine a pu les détecter.

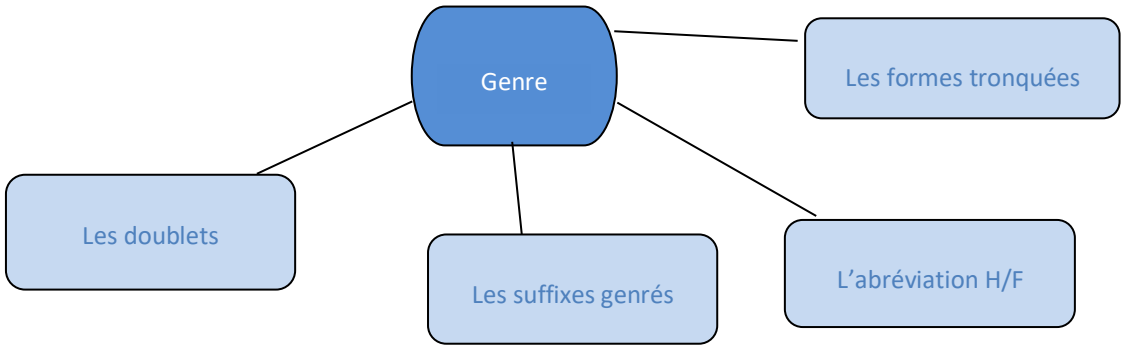
Comme points faibles, cette méthode prend beaucoup de temps si le corpus est large et donne des résultats limités si le corpus est petit et demande souvent une réannotation pour s'assurer du codage. Cependant, elle a pour point fort la bonne adaptation à l'étude des représentations sociales tels que les stéréotypes genrés dans notre cas.

2.2. Analyse quantitative

Nous avons mené une analyse quantitative de notre corpus qui consiste à faire des mesures systématiques de la parité et de l'inégalité dans les offres d'emploi et donc la place occupée par le genre féminin et masculin dans le contenu des offres. Notre méthode consiste à repérer des indicateurs de parité et de biais dans le titre et la description des offres d'emploi. Nous avons, tout d'abord, créé un premier modèle qui prend en compte le stéréotype de chaque métier. Les résultats de ce dernier n'étaient pas satisfaisants, nous avons donc effectué un travail d'adaptation et d'amélioration jusqu'à créer un deuxième modèle qui lui ne prend pas en compte le stéréotype mais le caractère épicène du métier. Vous trouverez cependant le premier modèle en ligne sur notre répertoire Github.

Quatre indices ont été pris en compte pour décider automatiquement la parité ou le biais du genre :

a. La mesure du niveau de parité qui repose sur (fig. 3) :
   - La détection des formes tronquées.
   - La détection des doublets dans les noms de métiers, les adjectifs, les articles définis et indéfinis et le troisième pronom personnel du singulier.
   - La détection des abréviation H/F.
b. La mesure de niveau de biais qui repose sur :
   - La détection des adjectifs et des noms de métiers genrés grâce à la détection du suffixe féminin seulement ou masculin seulement.



(Fig. 3)

A la fin de l'analyse, nous aurons, pour chaque annonce, deux valeurs de biais ; une pour le titre et une pour la description, qui varie entre 1 biaisé femme, 0 paritaire et -1 biaisé homme.

Nous calculons la moyenne des deux valeurs pour avoir cinq indices de biais tout comme l'analyse qualitative.

Comme toute méthode, l'analyse quantitative reposant sur l'extraction automatique d'information à l'aide des expressions régulières a des point forts et d'autres faibles. Dans notre étude, elle nous a permis de détecter rapidement la parité ou le biais du genre dans les annonces. Néanmoins, les modèles peuvent se tromper, détecter de fausses informations ou ne pas détecter la bonne, d'où vient l'importance de l'analyse qualitative.

Les différentes analyses que nous avons menées sont complémentaires et nous ont permis de décortiquer les différentes tendances d'usage des règles d'écriture épicène ainsi que la mesure de l'inégalité du genre dans les annonces d'emploi canadiens.

3. Les outils

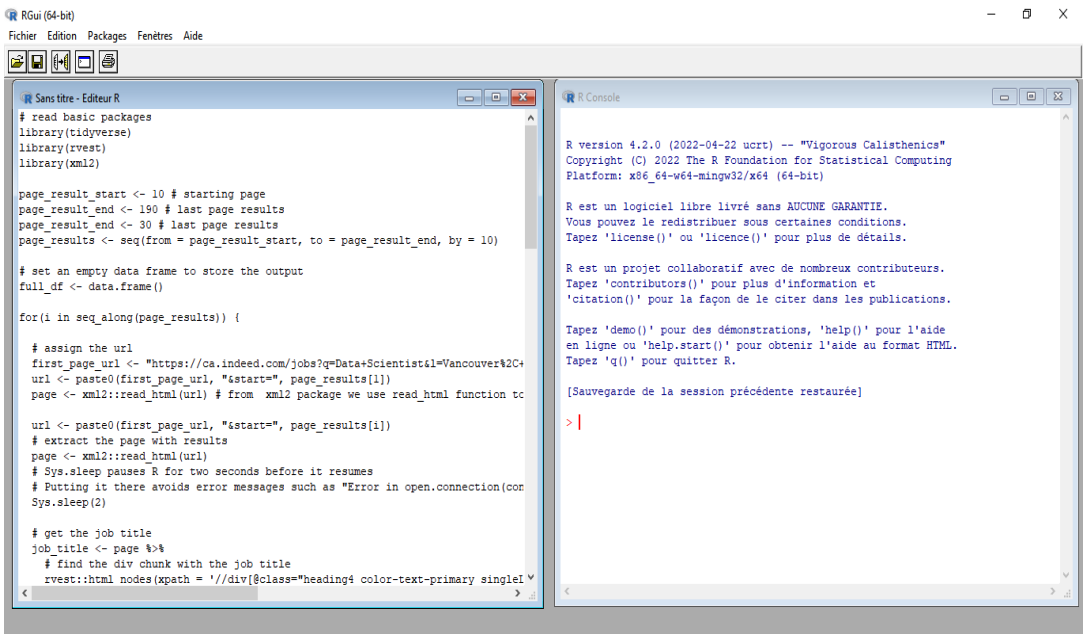
Les langages de programmation sont des langages informatiques créés par l'homme afin d'écrire un code source qui sera traité par la machine ensuite afin d'exécuter une commande. Ce terme est apparu pour la première fois dans les années cinquante et n'a cessé de se développer. A nos jours, nous assistons à une avancée technologique grâce à ses langages. En linguistique informatique ou en TAL, ces langages sont considérés comme des outils qui permettent le traitement du langage naturel humain par les machines. Le langage de programmation le plus utilisé en TAL aujourd'hui est Python.

Pour traiter et analyser notre corpus nous avons choisi de faire recours à deux langages de programmation R et Python. Les deux langages ont été peu complémentaires, l'usage d'un

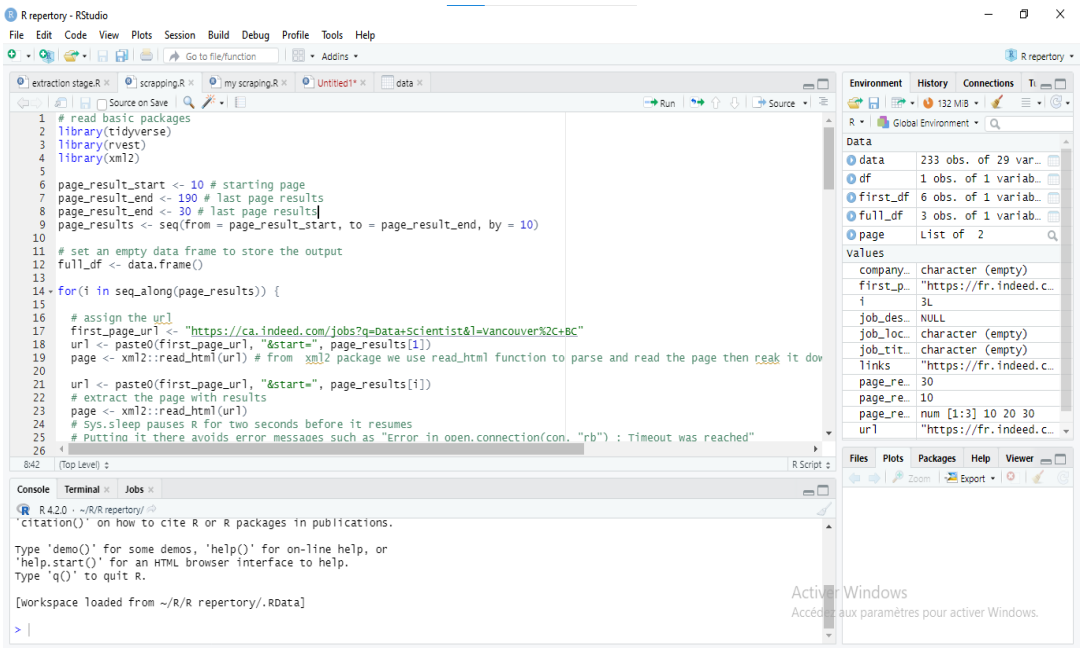
d'entre eux aurait été suffisant dans notre cas, mais l'objectif était de s'entrainer à coder et voir quel langage serait le plus avantageux. R et Python nous ont offert des avantages différents que nous allons traiter par la suite. Afin de collaborer avec notre tuteur sur ce projet, partager et héberger notre code nous avons utilisé Github.

3.1. R

R est un langage de programmation et un environnement très utilisé en statistique et en *Data Science*, développé par le laboratoire Bell aux États-Unis, il est téléchargeable sur : www.r-project.org . Sa spécificité consiste en la qualité des représentations graphiques qu'il offre avec des bibliothèques comme ggplot2 et webr visuellement et esthétiquement attrayantes et qui le différencie des autres langages. Il est aussi un langage de programmation *open-source* ce qui permet aux développeurs de contribuer à son développement et à personnaliser et développer des nouveaux packages. Cependant, R est plus lent en ce qui concerne le traitement des données massives et utilise plus de mémoire par rapport à d'autres langages tels que Python. R peut être utilisé directement sur son interface graphique de base (fig. 4), nous avons utilisé l'interface RStudio téléchargeable sur : https://www.rstudio.com (fig. 5) qui nous a facilité l'écriture des scripts en proposant la syntaxe avec la touche de tabulation. Il est aussi possible de visualiser et d'interagir avec les objets stockés dans l'environnement.



(Fig. 4)



(Fig. 5)

Les notions de base et les principaux objets

Certes, chaque langage de programmation a sa propre syntaxe qui le différencie et tout dépend de son objectif principal. Il peut avoir des notions ou des objets propres à lui, cependant, les langages partagent principalement les mêmes objets et notions de base, le cas de R et Python par exemple. Une bonne connaissance des différentes notions est indispensable pour écrire un code source correct.

Le jeu de données (data set) : est un ensemble de valeurs. Des jeux de données sont par défaut dans R tels que le jeu de données iris ou nasa.

La valeur (value) : est l'unité de base des données du langage R. Une valeur est de trois types (numérique, chaine de caractère ou logique). Des valeurs particulières peuvent être :

- NA (*not available*) : valeur manquante.
- NaN (*not a number*) : pas un nombre.
- -Inf, Inf : infini positif ou négatif.

La fonction (function) : c'est un bout de code qui accomplit une tâche spécifique. Une fonction permet de prendre les données, décomposer le problème en petits morceaux, et renvoyer un résultat. Les fonctions font partie de librairies ou modules mais nous pouvons créer nos propres fonctions si besoin.

Le vecteur (vector) : c'est une suite de valeurs unidimensionnelle et ordonnée pouvant être de type numérique, chaine de caractère ou logique TRUE ou FALSE. Sur R, nous pouvons tester la structure ou la nature d'un vecteur en utilisant la fonction str() ou class().

Le facteur (factor) : est un vecteur qui inclue un stock de valeurs d'une variable catégorielle de différents niveaux. Pour représenter les données, des fonctions R se réservent le traitement des facteurs tels que les fonctions d'analyse statistique comme plot(). Pour créer un facteur, il suffit de définir un vecteur et le transformer ensuite en utilisant la fonction factor(), pour extraire ses niveaux, il suffit d'utiliser la fonction level().

La matrice (matrix) : Un objet matrix est un tableau bi-dimensionnelle dont les données stockées sont d'un seul et même type. Pour créer une matrice, il faut utiliser la fonction matrix().

La liste (list) : est utilisé pour stocker différents éléments ou un ensemble d'objets de tailles et de natures diverses (vecteurs, matrices ou même des listes). C'est donc un tableau avec des données pouvant être hétérogènes. Pour créer une liste, utilisez la fonction list().

Le tableau (data frame) : est un tableau de données sous forme de vecteurs de même longueur ou taille à la différence de la liste mais pouvant être de différentes natures : valeurs numériques, alphanumériques ou booléennes contrairement à la matrice. Ce qui forme un tableau de type colonnes x lignes couramment utilisé dans R. Pour créer un tableau, il suffit d'utiliser la fonction data.frame() après avoir créé les vecteurs.

La boucle (Loop) : c'est un concept commun à tous les langages de programmation. Une boucle est une structure qui permet de répéter une instruction ou un ensemble d'instructions. C'est un ensemble de code à base d'une condition évalué vraie ou fausse. Tant que la condition est vraie, l'exécution du code se répète sinon elle s'arrête. Il y a principalement deux types de boucles

- Les boucles Pour 'for' : se veut une répétition d'une instruction un certain nombre de fois.
- Les boucles Tant que 'while' : sert à répéter une instruction sur la base d'une condition booléenne.

L'expression (statement) : c'est l'ensemble des fonctions, des variables et d'opérateurs. Elle peut être simple comme complexe.

L'expression conditionnelle (conditional statement) : c'est une expression qui permet l'exécution d'une commande en fonction de la condition. Nous repérons trois type de condition :

- if (condition) {expression 1 } else {expression 2 } : si la condition est vraie, R exécute le block de code dans l'expression if, si la condition est fausse, R exécute le block de code dans l'expression else.
- if (condition1) { expression 1} else if (condition2) {expression 2} else {expression 4} : avec l'expression else if ou elif nous pouvons ajouter des conditions autant qu'on veut.
- ifelse (ma condition, action si vrai, action si faux) : ifelse() est une fonction qui permet le test de l'expression, elle retourne de vraie ou faux.

Les fonctions de base

La fonction est un objet tous comme les autres objets déjà mentionnés. Une fonction peut prendre une autre fonction comme argument. R nous viens avec un ensemble de fonctions de bases dites intégrés ou *built-in functions* en anglais qui facilite la manipulation de nos données :

Fonction	Description
setwd()	Changement de répertoire
getwd()	Affichage du répertoire
install.package()	Installation de librairie depuis CRAN
installed.packages()	Affichage des librairie installés
install_github()	Installation de librairie depuis Github
library()	Chargement de librairie
help() ou ?	Accès à la documentation
class()	Donner la classe de l'objet
head()	Renvoie des premières lignes de données
view()	Visualisation de données
data()	Consulter des données
print()	Impression de l'argument à l'écran.
as.data.frame()	Création d'un data frame
as.character()	Conversion des données numériques à des chaines de caractères
seq()	Création d'une séquence d'éléments
colnames()	Nomination d'une colonne
file.choose()	Choisir un fichier
plot()	Production de graphiques de base
prcomp()	Retourner un objet de classe Prcomp
sys.sleep()	Suspension de l'exécution du code

paste0()	Concaténation des éléments sans séparateur
c()	Création ou combinaison de vecteurs
read.csv	Transformation du fichier .csv en data frame
write.csv()	Ecriture des fichiers .csv
row.names()	Nommer les lignes
fileEncoding	Définition du codant d'un fichier particulier
ncol()	Retourner le nombre de colonnes
gsub()	Remplacement des occurrences dans un string
round()	Arrondissement des valeurs numériques
rowMeans()	Calcul de moyenne de chaque ligne
factor()	Conversion une variable en facteur
Table()	Création d'un tableau
group_by()	Grouper des colonnes
ungroup()	Dissocier des colonnes
Ifelse()	Forme alternative et abrégée de l'instruction R if-else
nrow()	Renvoyer le nombre de lignes présentes dans un tableau.

3.1.1. Les opérateurs

Un opérateur est un élément de langage qui permet différentes opérations. Les éléments sur lesquels il s'applique sont appelés opérandes. Les opérateurs peuvent être unitaires donc ils s'appliquent sur un seul opérande ou binaires et demandent deux opérandes. R comprend plusieurs types d'opérateurs : arithmétiques, logiques, relationnels, d'affectation et d'autres opérateurs divers. Nous les présentons dans les tableaux suivants :

Les opérateurs d'affectation

Les opérateurs d'affectation sont des opérateurs de base qui permettent d'affecter une valeur à une variable.

Opérateurs d'affectation	Signification
=	Affectation de valeur
<-	Affectation de valeur
<<-	Affectation de valeur

Les opérateurs logiques

Sur la base d'une opération logique, ces opérateurs permettent le test des variables pour donner une réponse vraie 'True' ou 'fausse 'False'.

Opérateurs logiques	Signification
|	Ou inclusif : renvoie True si l'un des opérandes est vrai.
||	Ou inclusif : renvoie True si l'un des premiers opérandes est vrai.
&	Et : renvoie True si les deux opérandes sont vrai.
&&	Et : renvoie True si les deux premiers opérandes sont vrai.
!	Négation : inverser la valeur

Les opérateurs relationnels

Les opérateurs relationnels ou de comparaison permettent de comparer des éléments de type différents entre elles et sortir une valeur True ou False.

Opérateurs relationnels	Signification
<	Inferieur à
>	Supérieur à
<=	Inférieur ou égal à
>=	Supérieur ou égal à
==	Egale à
!=	Différent de ou inégal à

Les opérateurs arithmétiques

Semblable aux opérations mathématiques, ces opérateurs effectuent des opérations de calcul sur des opérandes de type numérique.

Opérateur arithmétique	Signification
+	Addition
-	Soustraction
*	Multiplication
/	Division
^	Exposant
**	Exposant
%/%	Division entière

%%	Modulo

Les opérateurs divers

Parmi les opérateurs divers que comprend R, nous avons utilisé les suivants :

Opérateurs divers	Signification
:	Impression d'une séquence
%>%	Renvoie de la valeur à l'expression suivante
$	Extraction d'un élément
%in%	Identifier si un élément appartient à un vecteur

Autres symboles

Ces symboles sont très fréquemment utilisés dans les différents langages de programmation :

Symbole	Signification
{ }	Agencement de dictionnaire
( )	Argument de fonction
[ ]	Agencement de liste
#	Ajout de commentaire

3.1.2. Les librairies utilisées

Les librairies ou les packages sont l'unité de base d'un langage de programmation. Elle comporte des fonctions regroupées sous un seul module pour permettre un usage spécifique. Chaque librairie vient avec sa propre documentation. R permet la contribution dans le développement des librairies comme déjà mentionné. Pour notre travail, nous avons fait recours à différentes librairies :

La librairie Tidyverse

Tidyverse est une vaste librairie qui comprend plusieurs extensions : stringr, ggplot, tidyr dplyr, purrr, readr, forcats et tibble. Elle permet donc un large nombre d'opérations dans

R tels que l'import/export de données, la manipulation des tableaux de données et de variables et leur visualisions ainsi que la programmation :

Fonction	Description
mutate()	Afficher la structure de l'objet de manière compact
select ()	Compter le nombre des patterns
filter()	Détecter un match dans un pattern
contains()	Sélection des variables qui correspondent au modèle

La librairie Stringr

La librairie Stringr comme son nom l'indique est utilisée pour faciliter la manipulation et le traitement des chaines de caractères appelés *strings* en anglais. Elle permet la préparation et le nettoyage des données ou le *Data Cleaning*, une des étapes de traitement de données textuelles. Toutes les fonctions de Stringr commencent par str_ :

Fonction	Description
Str()	Affichage de structure de l'objet de manière compact
str_count (x, pattern)	Compte de nombre des patterns
str_detect (x, pattern)	Détection d'un match dans un pattern
str_extract(x, pattern)	Extraction d'un pattern dans un string
str_subset (x, pattern)	Chercher la position d'un pattern
str_replace(x, pattern, replacement)	Remplacement de match par un nouveau dans le texte
str_replace_all (string, pattern, replacement)	Remplacement de toutes les occurrences d'un pattern
str_split()	Diviser la chaîne par délimiteur

La librairie Stringi

Construit au-dessus de Stringr, Stringi partage une convention et un objectif similaire avec Stringr :

Fonction	Description
stri_trim_both()	Suppression des espaces au début et à la fin de la chaîne

La librairie Dyplr

La librairie Dyplr permet la facilitation de traitement et de manipulation des données de type data frame ou tibble :

Fonction	Description
slice()	Sélection des lignes du tableau selon leur position
pull()	Extraction d'une colonne sous forme d'un vecteur
across()	Application de la même transformation sur multiple colonnes
Case_when	Test de condition
row_number()	Attribuer un numéro à chaque ligne

La librairie Rvest

La librairie Rvest permet de parser et donc chercher, récupérer ou récolter des données dans une page web et l'exploiter ensuite sur R :

Fonction	Description
html_nodes(x, css, xpath)	Extraire des nœuds d'un fichier Html
html_attr()	Extraire un attribut d'un fichier Html
html_text()	Extraire un texte d'un fichier Html

La librairie Xml2

Xml2 est une librairie conçue pour parser le contenu des fichiers de type Xml et Html :

Fonction	Description
read_html()	Lecture d'un fichier Html

La librairie Ggplot2

Ggplot2 est une librairie qui permet la visualisation et l'exploration des données en détails. Elle comprend des extensions comme Ggforce, Ggfortify et GGalli qui lui fournissent des fonctionnalités manquantes :

Fonction	Description
Ggplot()	Initialisation d'un objet ggplot
aes()	Création de map et ajout de variables
geom_boxplot()	Visualisation de la distribution d'une variable continue
geom_violin()	Visualiser la distribution des données et sa densité de probabilité
geom_smooth()	Ajout des moyens conditionnels lissés / ligne de regression
geom_point()	Création d'un nuage de points

geom_mark_ellipse()	Annotation des ensembles de points via des ellipses
autoplot()	Visualisation de divers objets
geom_bar()	Représentation des données en diagramme à barres
geom_text()	Ajouter du text à un graphique
theme()	Contrôle des éléments graphiques
scale_fill_viridis_d()	Fournir des cartes de couleurs uniformes
position_jitter()	Alignement des points générés par geom_point()
ggpairs()	Représentation des distributions des variables et leur corrélation

La libraire Weber

Nous avons fait recours à cette librairie pour son graphique circulaire visuellement attrayant :

Fonction	Description
PieDonut()	Dessiner un graphique circulaire

La librairie party

La librairie party est considéré comme une boîte à outils de calcul pour le partitionnement récursif. Elle permet la construction d'arbres de décision :

Fonction	Description
ctree()	Création d'arbre d'inférence conditionnelle
ctree_control()	Contrôle des arbres d'inférence conditionnelle
predict()	Prédiction des valeurs en fonction des données d'entrée.

Les expressions régulières

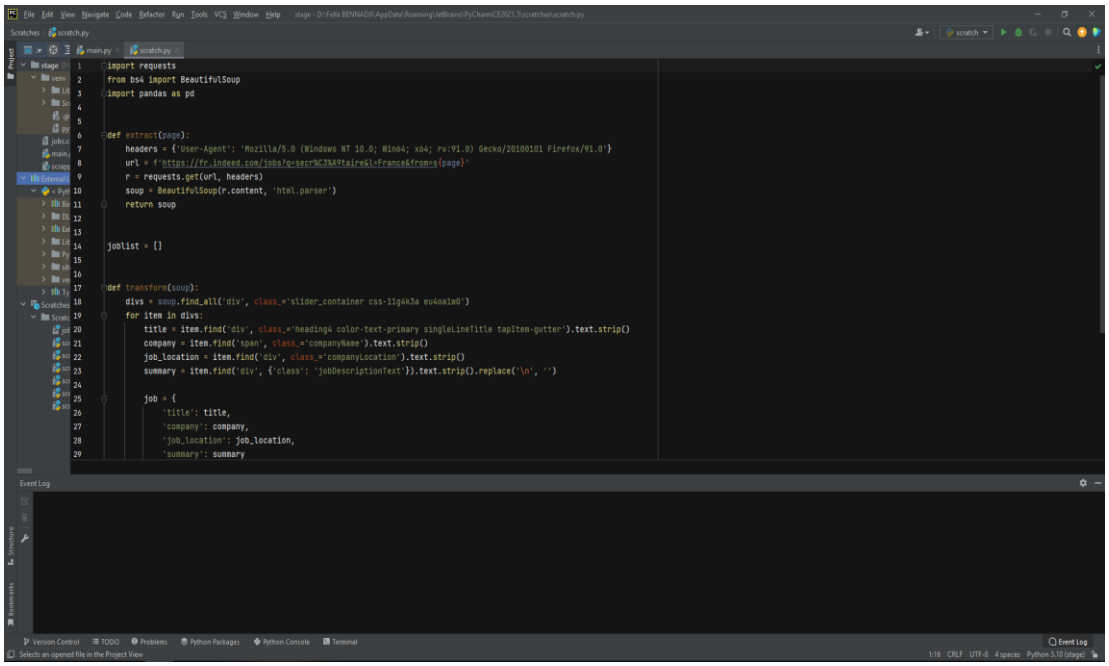
Les expressions régulières dit *Regex* servent à décrire avec concision un modèle de chaine de caractères pour ensuite le détecter ou l'extraire. Pour décrire un pattern, nous avons besoin des fonctionnalités clés des expressions régulières de stringr. Parmi ces fonctionnalités, nous trouvons :

Caractère	Signification
[:blank:]	Espace ou tabulation
[:space:]	Espace
[:alnum:]	Lettres et numéros

[:digit:]	Numéros
[:alpha:]	Lettres
[:punct:]	Ponctuation
[]	Créer une classe de caractères
\b	Correspond à la fin mot
\w	Correspond à un mot
\W	Correspond à tout caractère sauf le mot
.	N'importe quel caractère sauf nouvelle ligne
\	Caractère d'échappement
$	Correspond à la fin d'une chaine de caractère
^	Correspond au début d'une chaine de caractère
|	Ou
+	1 ou plus
*	0 ou plus
?	Facultatif

3.2. Python

Python est un langage de programmation orienté objet, populaire, très répandu auprès des développeurs web et de logiciels. Il a été développé par le programmeur néerlandais Guido van Rossum. La première version a vu le jour en février 1991 ensuite par l'association Python Software Foundation (PSF) à partir de la version Python 2.1 alpha. Pour ce travail, nous avons utilisé la version Python 3 téléchargeable sur : https://www.python.org avec l'interface Pycharm (fig. 6) : https://www.jetbrains.com/fr-fr/pycharm. Pycharm nous a permis une correction de la syntaxe et une mise en évidence des erreurs. Une autre alternative aurait été d'utiliser l'interface Spider ou Jupyter. Contrairement à R qui est destiné aux statistiques, Python est conçu principalement pour la programmation. Souvent utilisé pour développer des logiciels, il se différencie des autres langages de programmation de la famille C par sa syntaxe clair et simple. Considéré comme lisible, facile à utiliser et à apprendre, Python est très utilisé en TAL avec ses deux bibliothèques NLTK et SpaCy, ce qui justifie notre choix d'usage. Tout comme R python est gratuit et *open-source* avec une large communauté et une vaste documentation et forums en ligne. Python a l'avantage d'être plus rapide que R avec une meilleure performance cependant, ses librairies statistiques sont moins performantes.



(Fig. 6)

3.2.1. Les opérateurs

Tous comme R, Python comprend plusieurs types d'opérateurs très similaire à R avec quelques différences.

Les opérateurs d'affectation

Python a un seul opérateur d'affectation simple le = et d'autres opérateurs d'affectation dites composés qui permettent une opération de calcul suivie d'une affectation.

Opérateurs d'affectation	Signification
=	Affectation de valeur
+=	Addition puis affectation
-=	Soustraction puis affectation
*=	Multiplication puis affectation
/=	Division puis affectation
%=	Modulo puis affectation
//=	Division entière puis affectation
**=	Exponentiation puis affectation
&=	Et bit à bit puis affectation
|=	Ou bit à bit puis affectation
^=	XOR puis affectation
>>=	Décalage binaire à droite puis affectation
<<=	Décalage binaire à gauche puis affectation

Les opérateurs logiques

Ces opérateurs évaluent l'expression à 0 ou 1.

Opérateurs logiques	Signification
|	OU logique
^	OU exclusif logique
&	ET logique
~	NON logique
>>	Décalage binaire à droite
<<	Décalage binaire à gauche

Les opérateurs booléens

Contrairement aux opérateurs logiques, les opérateurs booléens évaluent l'expression True ou Flase.

Opérateurs booléens	Signification
or	OU booléen
and	ET booléen
not	NON booléen

Les opérateurs relationnels

Les opérateurs relationnels ou de comparaison permettent de comparer des éléments de type différents entre eux et de sortir une valeur True ou False.

Opérateurs relationnels	Signification
<	Strictement inferieur
>	Strictement supérieur
<=	Inférieur ou égal
>=	Supérieur ou égal
==	Egale
!=	Différent de ou inégal
<>	Différent

Les opérateurs d'identité

Les opérateurs d'identité se différencie des opérateurs relationnels par le fait de comparer les mêmes objets et leur emplacement mémoire.

Opérateurs d'identité	Signification
is	Représente le même objet
is not	Ne représente le même objet

Les opérateurs arithmétiques

Ces opérateurs ont la même fonction que les opérateurs arithmétiques du langage R.

Opérateur arithmétique	Signification
+	Addition
-	Soustraction
*	Multiplication
/	Division
**	Puissance
//	Division entière
%	Modulo

Les opérateurs d'appartenance

Ces opérateurs testent si la valeur spécifiée est présente dans l'objet de la séquence.

Opérateurs d'appartenance	Signification
in	Inclusion
not in	Non inclusion

Les fonctions de base

Tout comme R, L'interpréteur Python a un certain nombre de fonctions intégrées.

Fonction	Description
Print()	Impression de l'argument à l'écran.
append()	Ajout d'un seul élément à la liste existante
pip	Installation de librairie
install	Installation de librairie
Len()	Renvoie du nombre d'éléments d'un objet
split()	Division des chaîne de caractère en liste
help()	Accès à la documentation
open()	Ouverture d'un fichier

Les mots-clés

Les mots-clés sont indispensables pour l'écriture des scripts.

Mots-clés	Signification
as	Création d'un alias
from	Importation d'une section spécifiée d'un module.
import	Importation d'un module
class	Définition de classe de l'objet
def	Définition d'une fonction
for	Création d'une boucle
while	Création d'une boucle
in	Vérification de la présence d'une valeur
is	Test d'égalité de deux variables
return	Quitter une fonction et renvoyer une valeur
with	Simplifier la gestion des exceptions

3.2.2. Les librairies utilisées

Nous avons utilisé différentes librairies pour ce travail :

La librairie requests

Request est une librairie Python qui permet la gestion facile des requêtes http. A l'aide de ses fonctions, nous pouvons envoyer une requête au serveur pour nous renvoyer des données.

Fonction	Description
requests.get()	Envoyer d'une requête GET à un url spécifique

La librairie BeautifulSoup

Beautiful Soup est une librairie Python permettant d'extraire des données de fichiers HTML et XML et les transformer ensuite en une liste, un tableau ou dictionnaire Python. Elle est très utilisée en *Web Scraping* et Analyse du HTML.

Fonction	Description
BeautifulSoup()	Extraire du code HTML de la page
find_all()	Trouver toutes les balises avec le nom ou l'ID de balise spécifié
find()	Trouver la première balise de la valeur spécifiée.
get_text()	Renvoyer du texte dans la balise.

La librairie pandas

Souvent utilisée pour l'analyse et la manipulation de données de type tableaux 'DataFrames', pandas est une librairie Python très populaire, flexible et simple d'utilisation. Elle est généralement importée puis abrégé 'pd'

Fonction	Description
pd.DataFrame()	Création de data frame
transform()	Transformation des données

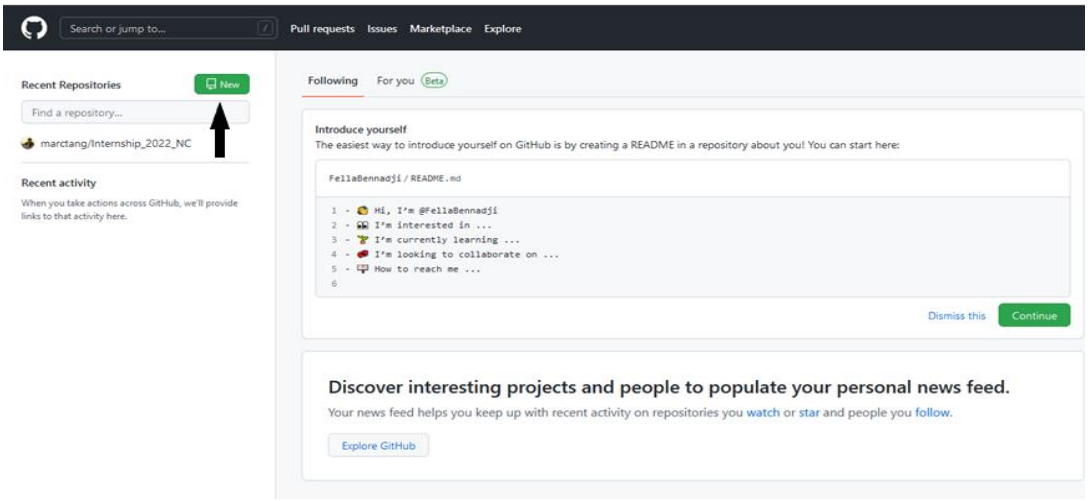
3.3. Github

Github est une plateforme d'hébergement de code très populaire auprès des personnes travaillant dans le domaine de technologie tels que les développeurs ou même les personnes qui font recours à la programmation ou le code pour des besoins précis comme les linguistes informaticiens. Lancé en Avril 2008 par des développeurs de logiciels, Github permet de d'héberger et de partager le code et de travailler ensemble sur des projets *open source*. Favorisant une approche collaborative, Il est considéré comme un réseau social qui facilite la création d'un profil personnel et le *personal branding* où toute personne pourrait visiter le profil et contribuer aux projets s'ils sont publics. Github est disponible gratuitement en ligne sur www.GitHub.com ou en version logiciel téléchargeable sur https://desktop.github.com . L'usage de Github est simple, il suffit de créer un compte sur la plateforme (fig. 7) :

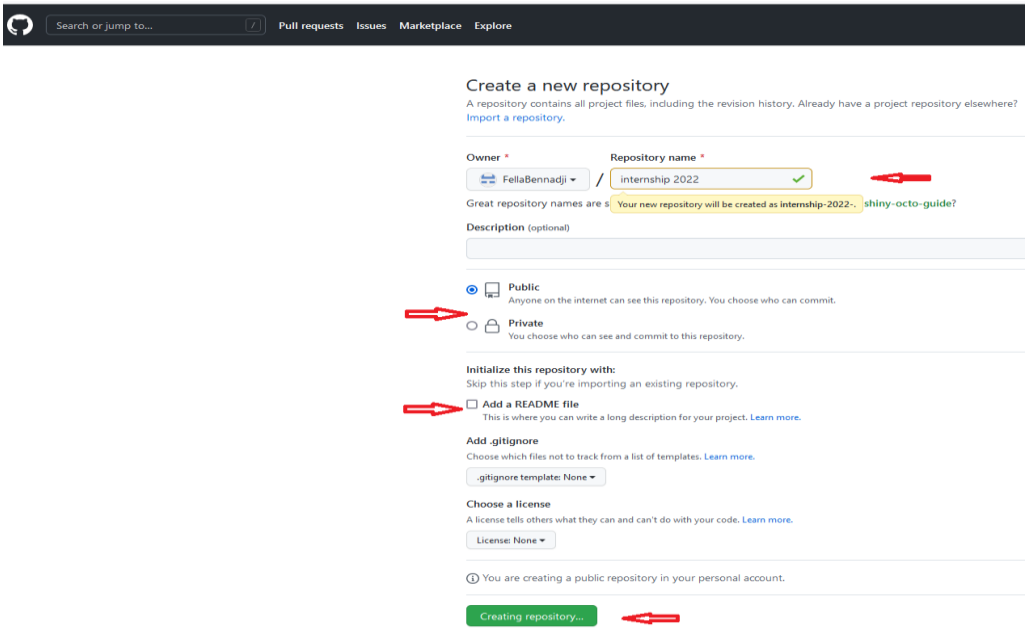
(Fig. 7)

Ensuite créer un référentiel Github, pour cela il suffit de cliquer sur nouveau (fig. 8) puis nommer le référentiel, ensuite choisir la confidentialité du référentiel public donc visible à tout

le monde ou privé. Une option serait d'ajouter un fichier README afin d'ajouter des informations sur ce référentiel puis cliquer sur créer un référentiel (fig. 9) :

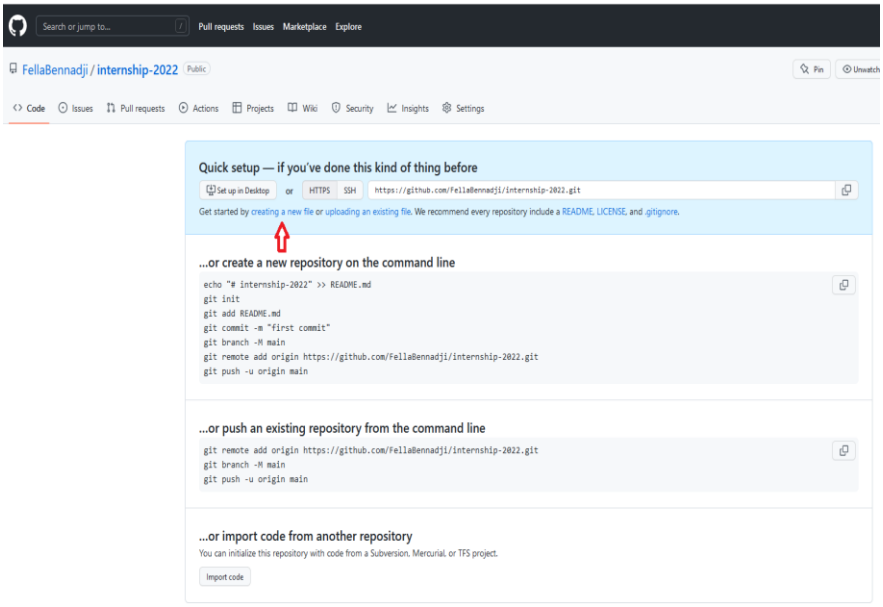


(Fig. 8)



(Fig. 9)

Pour ajouter des fichiers de différents types on peut cliquer sur ajouter un fichier (fig. 10).



(Fig. 10)

4. La programmation

Afin de collecter et analyser nos données, nous avons écrit notre code sur deux langages de programmation R et Python. Le code sera disponible prochainement à consulter sur cette adresse : https://github.com/marctang/Internship_2022_NC . Les étapes de code sont les mêmes, nous décrivons alors seulement le script R.

4.1. La collecte de données

Comme déjà mentionné (cf. 2. Data Scraping), trois étapes constituent le processus de la collecte de données : la sélection de la source de données, la collecte de données et enfin le stockage des données.

a. La sélection de la source de données

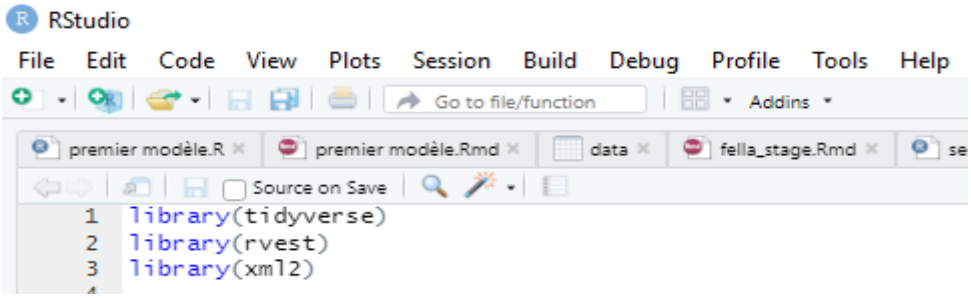
Pour collecter les données, il faut d'abord sélectionner la source de données ciblées. Nos données de source se situaient sur le web, plus précisément sur le site web Indeed Canada : https://.ca.indeed.com/.

b. Le grattage de données

La collecte des données dépend du type de données et de sa source qui va définir quelle technique de *Data Scraping* choisir. Puisque nos données se situent sur le web, nous avons donc

fait recours à la technique du *Web Scraping* afin de collecter des informations sur les deux métiers « secrétaire » et « chauffeur ».

Sur Rstudio, nous avons appelé les trois librairies : tidyverse pour manipuler nos données, xml2 pour parser le fichier html et rvest pour récupérer les données de la page web (fig. 11).



(Fig. 11)

Nous avons désigné la première page de recherche sur le site et la page de fin de recherche. Nous avons créé aussi une séquence qui comprend les deux pages de résultats (fig. 12)

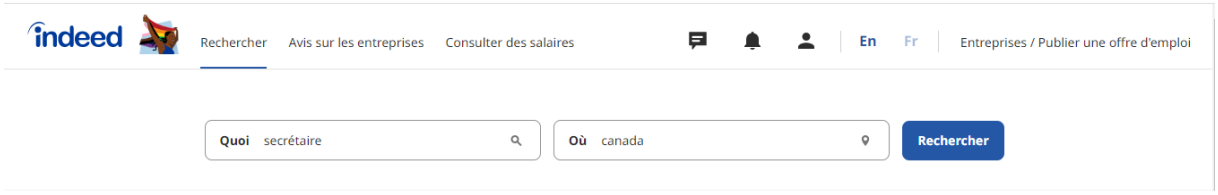
```
5 page_result_start <- 10 # starting page
6 page_result_end <- 100 # last page results
7 page_results <- seq(from = page_result_start, to = page_result_end, by = 10)
8
```

(Fig. 12)

Nous avons créé par la suite un tableau *data frame* et deux vecteurs, un pour les données des deux métiers (fig. 13) et un autre pour les liens que nous avons déjà cherchés sur le site (fig. 14).

```
9  # set an empty data frame to store the output
10 full_df <- NULL %>% data.frame()
11
12 # create a vector of the professions to look at
13 professions <- c("secretary",
14                  "driver")
15
16 # create a vector of the links
17 professions.links <- c("https://ca.indeed.com/jobs?q=secretary&l=Canada&vjk=5b09adbb54421f3d",
18                         "https://ca.indeed.com/jobs?q=driver&l=Canada&vjk=a7fb7b1cf75c2560")
19
```

(Fig. 13)



(Fig. 14)

Nous avons ouvert deux boucles de type pour ou for par la suite (fig. 15)

```
20  # loop for professions
21  for(z in 1:length(professions)){
22
23    # loop for page results
24    for(i in seq_along(page_results)) {
25
```

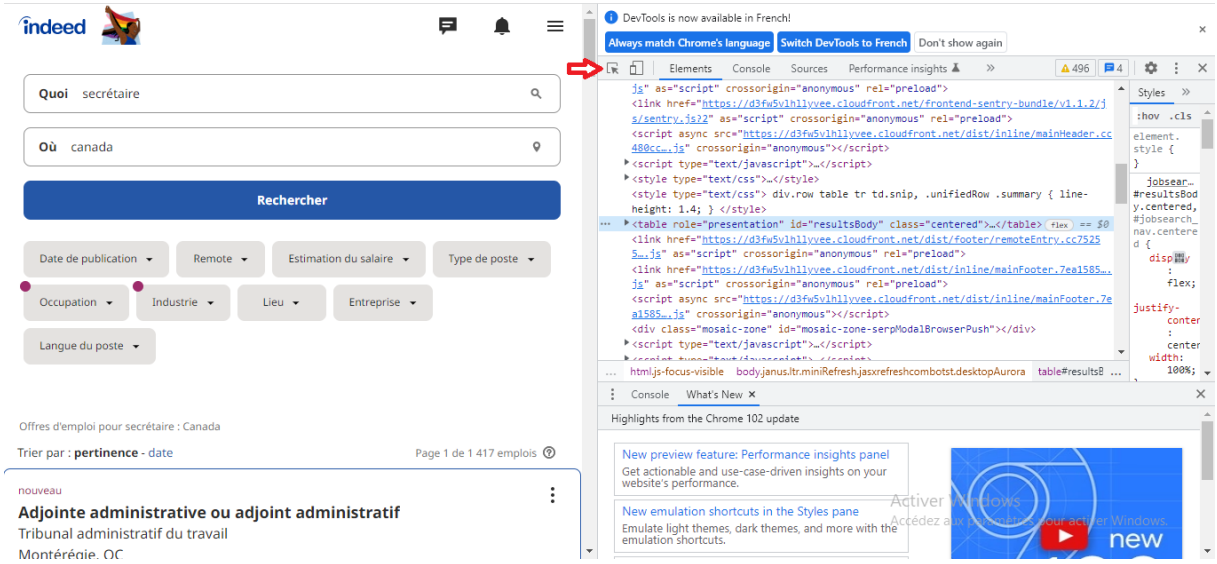
(Fig. 15)

Nous attribuons le lien à la variable url et nous utiliserons le package xml2 et la fonction read_html() pour lire et analyser et décomposer les différents éléments (, , <p>, etc.) de la page web. Nous mettons le système en veille pour deux secondes afin d'éviter les messages d'erreurs (fig. 16).

```
26    # assign the url
27    first_page_url <- professions.links[z]
28
29    url <- paste0(first_page_url, "&start=", page_results[i])
30    # extract the page with results
31    page <- xml2::read_html(url)
32    # Sys.sleep pauses R for two seconds before it resumes
33    # Putting it there avoids error messages such as "Error in open.connection(con, "rb") : Timeout was reached"
34    Sys.sleep(2)
35
```

(Fig. 16)

Nous envoyons une requête à l'aide de la librairie rvest au site web pour analyser le document HTML de celui-ci et extraire les différentes informations : l'intitulé du poste, le nom et l'emplacement de l'entreprise, le lien de l'annonce et la description du poste. Les éléments comme divs ( ), spans (), paragraphes (<p>) ou ancres (<a>) sont appelés des éléments nœuds *element nodes*, d'autres nœuds comprenant du texte sont appelés des nœuds d'attribut ou de texte *attribute or text nodes*. L'xpath quant à lui est un chemin pour extraire spécifiquement certaines parties d'un document arborescent tel que XML ou HTML. Pour retrouver ces différents éléments il faut aller inspecter le code source du site Web en cliquant droit et en sélectionnant inspecter. Ensuite, cliquez sur la petite flèche dans le coin supérieur gauche et survolez les éléments du site web d'Indeed. La petite flèche va sélectionner le code correspondant à l'élément que vous cherchiez (fig. 17).



(Fig. 17)

Après avoir repéré les différents éléments dans le document, nous avons effectué un test pour voir si notre code extrait exactement ce qu'on veut. Prenant l'exemple de l'intitulé du poste, la variable « job title » :

Nous avons examiné le code source et identifié l'intitulé de poste qui se situe dans les nœuds d'ancrage . Ensuite, nous avons examiné l'élément « span » et à partir de là, nous avons saisi l'attribut "Title" et extrait les informations. Des valeurs NA sont apparus, nous les avons enlevées par la suite. Le même processus s'est appliqué sur les autres différentes informations (fig. 18).

```
36    # get the job title
37    job_title <- page %>%
38      # find the div chunk with the job title
39      rvest::html_nodes(xpath = '//div[@class="heading4 color-text-primary singleLineTitle tapItem-gutter"]') %>%
40      # find the span chunk with the job title
41      rvest::html_nodes("span") %>%
42      # extract the attribute of title
43      rvest::html_attr("title")
44
45    # removing NAs
46    job_title <- job_title[!is.na(job_title)]
47
48
49    # get the company name
50    company_name <- page %>%
51      # extract the span chunk with company name
52      rvest::html_nodes(xpath = '//span[@class="companyName"]')  %>%
53      # extract its text
54      rvest::html_text() %>%
55      # remove white spaces on both sides
56      stringi::stri_trim_both()
57
58
59    # get job location
60    job_location <- page %>%
61      # extract div chunk with company location
62      rvest::html_nodes(xpath = '//div[@class="companyLocation"]')%>%
63      # extract its text
64      rvest::html_text() %>%
65      # remove white spaces on both sides
66      stringi::stri_trim_both()
```

(Fig. 18)

Sauf que pour extraire la description du poste, le simple fait de les extraire directement à partir des éléments de document HTML va nous extraire une petite méta description. Nous voudrions obtenir la description complète, il a donc fallu collecter d'abord les liens des offres qui se situent sous l'élément 'a', l'attribut 'href' et à partir de là extraire le texte de la description, enfin supprimer les espaces blancs avec la fonction stri_trim_both() et sauvegarder les liens dans un vecteur (fig. 19).

```
    # get links
    links <- page %>%
      # extract the anchors 'a' chunks
      rvest::html_nodes("a") %>%
      # extract the href link attributes
      html_attr("href") %>%
      # only keep the one annotated as rc or company
      str_subset("^/company|^/pagead|^/rc")

    # create an empty vector to store the descriptions
    job_description <- c()
    # for each link of the results
    for(x in seq_along(links)) {
      # get the page
      url <- paste0("https://ca.indeed.com", links[x])
      page <- xml2::read_html(url)
      #Sys.sleep(2)

      job_description[[x]] <- page %>%
        # find the div chunk with the job description
        rvest::html_nodes(xpath = '//div[@id="jobDescriptionText"]') %>%
        # extract its text
        rvest::html_text() %>%
        # remove white spaces on both sides
        stringi::stri_trim_both()
    }

    # save the links as a vector
    links <- paste0("https://ca.indeed.com", links)

```

(Fig. 19)

Après l'extraction de toutes les informations, nous les combinons dans le tableau à l'aide de la fonction cbind() et nous fermons les deux boucles que nous avons ouvertes au début (fig. 20).

```
    # combine the extracted information for the page of results
    df <- cbind(# adding which main job it is
      professions[z],
      # adding extracted info
      job_title, company_name, job_location, job_description, links) %>%
      as.data.frame()

    # add an if statement to avoid empty data
    if(ncol(df) == 6){
      # combine the results from a page of results with the other pages of results
      full_df <- rbind(full_df, df)
    }else{
      # do nothing if there is not data from that page
    }
  } # loop for page results

} # loop for professions

```

(Fig. 20)

Nous nommons toutes les colonnes à l'aide de la fonction colnames() et nous les ajustons et les intégrons à la data frame à l'aide de la fonction mutate() (fig. 21).

```
117 # adjust the column names
118 colnames(full_df) <- c("Profession","Title","Company","Location","Description","Links")
119 # adjust the character format
120 full_df <- full_df %>%
121   mutate(across(everything(), as.character))
122
```

(Fig. 21)

c. Le stockage des données

La conservation et le stockage des données se fait sur une base de données ou sous un fichier de type Excel, CSV, HTML ou autre. Nous vérifions notre tableau pour s'assurer de notre résultat à l'aide de la fonction view() puis nous sauvegardons le fichier sous format .csv. Le sigle CSV signifie *Comma-Separated Values*, format de texte simple qui utilise des virgules ou des points-virgules pour séparer des champs ou des colonnes de données d'une base de données. Une dernière étape est de choisir la répertoire d'encodage afin de permettre la bonne lecture des caractères par la suite (fig. 22).

```
124 # visual check
125 view(full_df)
126
127 # saving the output
128 full_df %>% write.csv("data_raw/Indeed_search_CA.csv",
129                       row.names = FALSE,
130                       fileEncoding = "UTF-8")
```

(Fig. 22)

4.2. La préparation de données

Après la collecte de données, nous avons analysé nos données quantitativement (cf. 2.2 Analyse quantitative). Afin de préparer nos données à l'analyse quantitative, nous importons notre tableau annoté qualitativement et nous suivons différentes étapes de préparation de données ou *Data-Wrangling*.

a. Le nettoyage de données

Après la collecte de données, l'étape qui suit est l'exploration et l'analyse. Pour explorer nos données, la première étape est le *Data Cleaning* ou le nettoyage de données. Comme les données peuvent être bruyantes, incomplètes ou contenir des erreurs, nous avons effectué un nettoyage manuel lors de l'analyse qualitative. En vérifiant notre tableau nous nous sommes rendu compte que notre code avait gratté quelques offres en anglais, d'autres offres dans le

domaine informatique. Nous avons donc nettoyé le tableau pour s'assurer d'avoir une analyse correcte du corpus et des résultats d'exploration de données exacts.

b. La sélection des données

Pour commencer l'analyse, nous appelons les différentes librairies nécessaires : tidyverse pour manipuler le tableau et stringi pour manipuler les données de type chaines de caractères. Nous téléchargeons le fichier csv créé dans la première partie. Toutes nos données intégrées ne sont évidemment pas nécessaires pour l'exploration de données, nous sélectionnons donc quelques données pour extraire uniquement les informations utiles de la grande base de données. Nous mettons les descriptions et les intitulés en minuscule pour préparer l'étape suivante et nous visualisons le tableau à l'aide de la fonction view() (fig. 23).

```
library(tidyverse)
library(stringr)

# Run the following code to get the raw qualitative data
data = read.csv(file.choose(), sep=";", dec=";", header=TRUE) %>%
  mutate(ID_job=row_number(),
       Title = tolower(Title),
       Description = tolower(Description)) %>%
  select(ID_job,Profession,Title,Title.value, Description, Description.value, X4.groups)
view(data)
```

(Fig. 23)

Avant de passer à l'étape suivante, nous calculons la moyenne de biais qualitative des offres en les variables du titre et de la description (cf. 2.1 Analyse qualitative) (fig. 24).

```
# For Qualitative result

# For "Qualitative result" column
data_R2 <- data %>%
  mutate(Qualitative.result = rowMeans(data[c('Title.value',
                                              'Description.value')]))
```

(Fig. 24)

c. L'extraction de données :

Une fois les données sélectionnées, nous pouvons commencer le processus d'extraction. Nous avons voulu détecter, extraire et compter les différents indicateurs de biais et de parité dans l'intitulé et la description du poste vu qu'il y a souvent un non alignement entre titre et description. Étant donné que nos données sont déjà préparés, nous utilisons directement la fonction Str_count() et str_detect() qui nous permettent de compter et détecter les indicateurs et la fonction mutate() pour les ajouter comme variable dans le tableau. Comme nos indicateurs

de genre sont des patterns, nous les avons décrites grâce aux expressions régulières '*Regex'*, un outil concis et flexible pour décrire des modèles dans des chaînes de caractères.

Dans un premier temps, le titre peut être un simple nom de métier comme « secrétaire » ou une désignation adjectivale (nom + adjectif) comme « agent administratif ». Vu sa limite de longueur, et pour la parité, nous avons compté et extrait les formes tronquées, les doublets (noms de métier et adjectif) et l'abréviation H/F. Pour le biais nous avons cherché la présence des suffixes des noms de métiers ou des désignations adjectivales genrées. Comme « secrétaire » est un nom de métier épicène, nous avons donc pris cela en compte dans notre analyse et nous avons créé une fonction qui le détecte (fig. 25).

```
#count occurances in "Title" column

#for truncated forms
data_R2 <- data_R2 %>%
  group_by(ID_job) %>%
  mutate(ve.count.tit = str_count(Title, "[:space:]?(?<=\\(|\\.)ve\\b(?=\\)|[:space:]|\\.|,|$)"),
                    ive.count.tit = str_count(Title, "[:space:]?(?<=\\(|\\.)ive\\b(?=\\)|[:space:]|\\.|,|$)"),
                    e.count.tit = str_count(Title, "[:space:]?(?<=\\(|\\.)e\\b(?=\\)|[:space:]|\\.|,|$)"),
                    trice.count.tit = str_count(Title, "[:space:]?(?<=\\(|\\.)trice\\b(?=\\)|[:space:]|\\.|,|$)"),
                    ère.count.tit = str_count(Title, "[:space:]?(?<=\\(|\\.)ère\\b(?=\\)|[:space:]|\\.|,|$)"),
                    euse.count.tit = str_count(Title, "[:space:]?(?<=\\(|\\.)euse\\b(?=\\)|[:space:]|\\.|,|$)"),
                    ne.count.tit = str_count(Title, "[:space:]?(?<=\\(|\\.)ne\\b(?=\\)|[:space:]|\\.|,|$)"),
                    sum_troncat_title = ve.count.tit + ive.count.tit + e.count.tit + trice.count.tit +
                                        ère.count.tit + euse.count.tit + ne.count.tit) %>%
  ungroup()

#for H/F
data_R2 <- data_R2 %>%
  group_by(ID_job) %>%
  mutate( hf.count.tit = str_count(Title, "h/f"),
          sum_hf_title = hf.count.tit) %>%
  ungroup()

#for doublet
data_R2 <- data_R2 %>%
  group_by(ID_job) %>%
  mutate( if.ive.tit = str_detect(Title,'(?<!\\b)ive\\b') == TRUE & str_detect(Title,'(?<!\\b)if\\b') == TRUE,
          ien.ienne.tit = str_detect(Title,'(?<!\\b)ienne\\b') == TRUE & str_detect(Title,'(?<!\\b)ien\\b') == TRUE,
          teur.trice.tit = str_detect(Title,'(?<!\\b)trice\\b') == TRUE & str_detect(Title,'(?<!\\b)teur\\b') == TRUE,
          ier.ère.tit = str_detect(Title,'(?<!\\b)ère\\b') == TRUE & str_detect(Title,'(?<!\\b)i?er\\b') == TRUE,
          eur.euse.tit = str_detect(Title,'(?<!\\b)euse\\b') == TRUE & str_detect(Title,'(?<!\\b)eur\\b') == TRUE,
          al.ale.tit = str_detect(Title,'(?<!\\b)ale\\b') == TRUE & str_detect(Title,'(?<!\\b)al\\b') == TRUE,
          é.ée.tit = str_detect(Title,'(?<!\\b)ée\\b') == TRUE & str_detect(Title,'(?<!\\b)é\\b') == TRUE,
          sum_doublet_title = ifelse(TRUE %in% c(if.ive.tit, teur.trice.tit, ier.ère.tit, eur.euse.tit, al.ale.tit,
                                                 ien.ienne.tit, é.ée.tit),1,0)) %>%
  ungroup()

#for masculine adjective
data_R2 <- data_R2 %>%
  group_by(ID_job) %>%
  mutate( if.tit = str_detect(Title,'(?<!\\b)ive\\b') == FALSE & str_detect(Title,'(?<!\\b)if\\b') == TRUE,
          ien.tit = str_detect(Title,'(?<!\\b)ienne\\b') == FALSE & str_detect(Title,'(?<!\\b)ien\\b') == TRUE,
          teur.tit = str_detect(Title,'(?<!\\b)trice\\b') == FALSE & str_detect(Title,'(?<!\\b)teur\\b') == TRUE,
          ier.tit = str_detect(Title,'(?<!\\b)ère\\b') == FALSE & str_detect(Title,'(?<!\\b)i?er\\b') == TRUE,
          eur.tit = str_detect(Title,'(?<!\\b)euse\\b') == FALSE & str_detect(Title,'(?<!\\b)eur\\b') == TRUE,
          al.tit = str_detect(Title,'(?<!\\b)ale\\b') == FALSE & str_detect(Title,'(?<!\\b)al\\b') == TRUE,
          é.tit = str_detect(Title,'(?<!\\b)ée\\b') == FALSE & str_detect(Title,'(?<!\\b)é\\b') == TRUE,
          pres_masc = TRUE %in% c(if.tit, teur.tit ,ier.tit ,eur.tit ,al.tit ,ien.tit ,é.tit)) %>%
  ungroup()

#for feminine adjective
data_R2 <- data_R2 %>%
  group_by(ID_job) %>%
  mutate( ive.tit = str_detect(Title,'(?<!\\b)ive\\b') == TRUE & str_detect(Title,'(?<!\\b)if\\b') == FALSE,
          ienne.tit = str_detect(Title,'(?<!\\b)ienne\\b') == TRUE & str_detect(Title,'(?<!\\b)ien\\b') == FALSE,
          trice.tit = str_detect(Title,'(?<!\\b)trice\\b') == TRUE & str_detect(Title,'(?<!\\b)teur\\b') == FALSE,
          ère.tit = str_detect(Title,'(?<!\\b)ère\\b') == TRUE & str_detect(Title,'(?<!\\b)i?er\\b') == FALSE,
          euse.tit = str_detect(Title,'(?<!\\b)euse\\b') == TRUE & str_detect(Title,'(?<!\\b)eur\\b') == FALSE,
          ale.tit = str_detect(Title,'(?<!\\b)ale\\b') == TRUE & str_detect(Title,'(?<!\\b)al\\b') == FALSE,
          ée.tit = str_detect(Title,'(?<!\\b)ée\\b') == TRUE & str_detect(Title,'(?<!\\b)é\\b') == FALSE,
          pres_fem = TRUE %in% c(ive.tit, trice.tit ,ère.tit ,euse.tit ,ale.tit ,ienne.tit ,ée.tit)) %>%
  ungroup()

#for epicene title profession
data_R2 <- data_R2 %>%
  group_by(ID_job) %>%
  mutate(sum_epicene_title = ifelse(check_epicene_metier(Title) == TRUE,1,0)) %>%
  ungroup()
```

(Fig. 25)

Dans un second temps, nous avons détecté la parité dans la description à l'aide des formes tronquées, l'abréviation H/F, les doublets dans les noms de métier, les désignations adjectivales, les articles définis et indéfinis et les pronoms personnels de troisième personne du singulier, ainsi que la présence des quatre groupes. Cependant, et vu que nous faisons recours qu'au *Regex* et à cause de la longueur de texte de la description, la détection des indicateurs de biais (les noms de métiers et les adjectifs genrés) était impossible. La description peut donc identifier la parité mais non pas le biais (fig. 26).

```
#count occurances in "Description" colomn

#for truncated forms
data_R2 <- data_R2 %>%
  group_by(ID_job) %>%
  mutate(ve.count.desc = str_count(Description, "[:space:]?(?<=\\(|\\.)ve\\b(?=\\)|[:space:]|\\.|,|$)"),
         ive.count.desc  = str_count(Description , "[:space:]?(?<=\\(|\\.)ive\\b(?=\\)|[:space:]|\\.|,|$)"),
         e.count.desc  = str_count(Description , "[:space:]?(?<=\\(|\\.)e\\b(?=\\)|[:space:]|\\.|,|$)"),
         trice.count.desc  = str_count(Description , "[:space:]?(?<=\\(|\\.)trice\\b(?=\\)|[:space:]|\\.|,|$)"),
         ère.count.desc = str_count(Description , "[:space:]?(?<=\\(|\\.)ère\\b(?=\\)|[:space:]|\\.|,|$)"),
         euse.count.desc  = str_count(Description , "[:space:]?(?<=\\(|\\.)euse\\b(?=\\)|[:space:]|\\.|,|$)"),
         ne.count.desc  = str_count(Description , "[:space:]?(?<=\\(|\\.)ne\\b(?=\\)|[:space:]|\\.|,|$)"),
         sum_troncat_description = ve.count.desc  + ive.count.desc + e.count.desc + trice.count.desc  +
           ère.count.desc  + euse.count.desc  + ne.count.desc) %>%
  ungroup()

#for H/F
data_R2 <- data_R2 %>%
  group_by(ID_job) %>%
  mutate(hf.count.desc  = str_count(Description, "h/f"),
         sum_hf_description = hf.count.desc) %>%
  ungroup()

#for doublet
data_R2 <- data_R2 %>%
  group_by(ID_job) %>%
  mutate(eur.euse.desc = str_detect(Description,
                                    "((?<!\\b)eur(([:blank:](\\w+[:blank:])?ou([:blank:]\\w+)?[:blank:])|([:blank:]?[:punct:][:blank:]?))\\w+euse\\b)|
                                    (((?<!\\b)euse(([:blank:](\\w+[:blank:])?ou([:blank:]\\w+)?[:blank:])|([:blank:]?[:punct:][:blank:]?))\\w+eurs?\\b))"),
         if.ive.desc = str_detect(Description,
                                  "((?<!\\b)if(([:blank:](\\w+[:blank:])?ou([:blank:]\\w+)?[:blank:])|([:blank:]?[:punct:][:blank:]?))\\w+ive\\b)|
                                  (((?<!\\b)ive(([:blank:](\\w+[:blank:])?ou([:blank:]\\w+)?[:blank:])|([:blank:]?[:punct:][:blank:]?))\\w+if\\b))"),
         teur.trice.desc = str_detect(Description,
                                      "((?<!\\b)teur(([:blank:](\\w+[:blank:])?ou([:blank:]\\w+)?[:blank:])|([:blank:]?[:punct:][:blank:]?))\\w+trice\\b)|
                                      (((?<!\\b)trice(([:blank:](\\w+[:blank:])?ou([:blank:]\\w+)?[:blank:])|([:blank:]?[:punct:][:blank:]?))\\w+teur\\b))"),
         teur.trice.desc = str_detect(Description,
                                      "((?<!\\b)teur(([:blank:](\\w+[:blank:])?ou([:blank:]\\w+)?[:blank:])|([:blank:]?[:punct:][:blank:]?))\\w+trice\\b)|
                                      (((?<!\\b)trice(([:blank:](\\w+[:blank:])?ou([:blank:]\\w+)?[:blank:])|([:blank:]?[:punct:][:blank:]?))\\w+teur\\b))"),
         ien.ienne.desc = str_detect(Description,
                                     "((?<!\\b)ien(([:blank:](\\w+[:blank:])?ou([:blank:]\\w+)?[:blank:])|([:blank:]?[:punct:][:blank:]?))\\w+ienne\\b)|
                                     (((?<!\\b)ienne(([:blank:](\\w+[:blank:])?ou([:blank:]\\w+)?[:blank:])|([:blank:]?[:punct:][:blank:]?))\\w+ien\\b))"),
         al.ale.desc = str_detect(Description,
                                  "((?<!\\b)al(([:blank:](\\w+[:blank:])?ou([:blank:]\\w+)?[:blank:])|([:blank:]?[:punct:][:blank:]?))\\w+ale\\b)|
                                  (((?<!\\b)ale(([:blank:](\\w+[:blank:])?ou([:blank:]\\w+)?[:blank:])|([:blank:]?[:punct:][:blank:]?))\\w+al\\b))"),
         nt.nte.desc = str_detect(Description,
                                  "((?<!\\b)nt(([:blank:](\\w+[:blank:])?ou([:blank:]\\w+)?[:blank:])|([:blank:]?[:punct:][:blank:]?))\\w+nte\\b)|
                                  (((?<!\\b)nte(([:blank:](\\w+[:blank:])?ou([:blank:]\\w+)?[:blank:])|([:blank:]?[:punct:][:blank:]?))\\w+nt\\b))"),
         le.la.desc = str_detect(Description,
                                 "\\ble(([:blank:](\\w+[:blank:])?ou([:blank:]\\w+)?[:blank:])|([:blank:]?[:punct:][:blank:]?))la\\b|
                                 \\bla(([:blank:](\\w+[:blank:])?ou([:blank:]\\w+)?[:blank:])|([:blank:]?[:punct:][:blank:]?))le\\b"),
         un.une.desc = str_detect(Description,
                                  "\\bun(([:blank:](\\w+[:blank:])?ou([:blank:]\\w+)?[:blank:])|([:blank:]?[:punct:][:blank:]?))une\\b|
                                  bune(([:blank:](\\w+[:blank:])?ou([:blank:]\\w+)?[:blank:])|([:blank:]?[:punct:][:blank:]?))un\\b"),
         il.elle.desc = str_detect(Description,
                                   "\\bil(([:blank:](\\w+[:blank:])?ou([:blank:]\\w+)?[:blank:])|([:blank:]?[:punct:][:blank:]?))elle\\b|
                                   \\belle(([:blank:](\\w+[:blank:])?ou([:blank:]\\w+)?[:blank:])|([:blank:]?[:punct:][:blank:]?))il\\b"),

         sum_doublet_description = ifelse(TRUE %in% c(if.ive.desc, teur.trice.desc, eur.euse.desc ,al.ale.desc ,
                                                      ien.ienne.desc, nt.nte.desc, le.la.desc, un.une.desc, il.elle.desc),1,0)) %>%
  ungroup()

#for 4 groups
data_R2 <- data_R2 %>%
  group_by(ID_job) %>%
  mutate(X4.groups.description = str_count(Description, "(programme|égalité).?d.?accès.?à.?l.?(égalité|emploi)")) %>%
  ungroup()
```

(Fig. 26)

Pour s'assurer de l'exactitude de notre extraction, nous avons testé nos expressions régulières. Selon les résultats de test, notre *Regex*, arrive à détecter les formes tronquées mais avec quelques bruits causés par l'extraction de la forme (e) venant de quelques abréviations présentes dans les textes. Comme les offres d'emploi sont canadiens, nous avons rencontré des annonces bilingues ce qui a aussi causé du bruit. Les *Regex* arrive à extraire 82.7 % des mentions du programme d'égalité d'accès à l'emploi, nous avons donc 17.3% des mentions de programme

d'égalité qui n'ont pas été détectées, ceci est dû à la limite de notre méthode de détection par règles (fig. 27).

```
#TEST
# we need to do random checking to see if the functions capture what we want
str_detect(c("agent(e)",
             "agent.e ",
             "administratif(ve)",
             "conduc(trice)",
             "conduc.trice ",
             "administratif.ve ",
             "administrat.ive",
             "journali(ère)",
             "livreur(euse)",
             "mécanicien(ne)"),
           "[\\(\\.]ve[\\) \\.\\,]",
           "[\\(\\.]ive[\\) \\.\\,]")


# or look at the text of the 1st row and extract the count that we generated
data_R2 %>%
  # take a specific row number
  slice(8) %>%
  # show the description
  pull(Description)

# extract the counts of items and compare with the previous text
data_R2 %>%
  # take a specific row number
  slice(8) %>%
  # take relevant columns
  select(e.count.tit)

#test also 4 groups
data_R2 %>%
  filter(X4.groups == X4.groups.description) %>%
  select(Profession, X4.groups, X4.groups.description)

taux_exact <- nrow(data_R2 %>%
                     filter((X4.groups == 0 & X4.groups.description != 0) |
                              (X4.groups == -1 & X4.groups.description == 0)))/nrow(data_R)*100
taux_exact
```

(Fig. 27)

Nous voudrons maintenant mesurer le biais à l'aide des nouvelles données extraites en utilisant la fonction ifelse() (fig. 28).

Pour quantifier la parité ou l'inégalité du genre dans le titre de l'annonce, nous avons demandé à la machine de regarder d'abord dans le titre qui comprend un nom de métier épicène, puis voir s'il y a une présence d'indicateur de parité (forme tronquée, doublet ou abréviation h/f), ensuite regarder les indicateurs de biais (suffixe masculin ou féminin). La distinction nom épicène ou genré est dû au fait qu'un titre peut être long et comprendre des éléments qui peuvent falsifier le résultat obtenu.

Cependant, pour la description, nous n'avons pu quantifier que la parité. En cas d'absence d'indicateur de parité, nous avons demandé à la machine de se référer au résultat du titre pour décider.

La méthode appliquée à la description a l'inconvénient de conduire la machine à coder la description fausse en cas de biais ou de parité exprimée autrement (les formes épicènes comme les pronoms personnels épicènes comme vous et les mots épicènes comme la personne) car une description peut ne pas suivre le titre ; un titre peut être paritaire tandis que la description est inégalitaire par exemple.

Notre indice de biais se trouve dans une nouvelle colonne nommée Bias.indicator qui comprend la moyenne des deux variables titre et description. Nous testons le résultat quantitatif en le comparant avec le résultat qualitatif pour avoir un taux de précision.

```
#Analysis

#for Quantitative title
data_R2 <- data_R2 %>%
  mutate(Quanti.title = ifelse(sum_epicene_title > 0 & sum_hf_title + sum_doublet_title + sum_troncat_title > 0,
                        0,
                        ifelse(sum_epicene_title > 0 & pres_masc == TRUE,
                               -1,
                               ifelse(sum_epicene_title > 0 & pres_fem == TRUE,
                                      1,
                                      ifelse(sum_epicene_title == 0  & sum_troncat_title > 0,
                                             0,
                                             ifelse(sum_epicene_title == 0 & sum_hf_title > 0,
                                                    0,
                                                    ifelse(sum_epicene_title == 0 & sum_doublet_title > 0,
                                                           0,
                                                           ifelse(sum_epicene_title == 0 & pres_masc == TRUE,
                                                                  -1,
                                                                  ifelse(sum_epicene_title == 0 & pres_fem == TRUE,
                                                                         1,
                                                                         0)))))))

  ))

#test
data_R2 %>%
select(Profession,Title, Title.value,Quanti.title) %>%
  filter(Title.value != Quanti.title) %>%
  select(Profession) %>% table()

taux_exact <- nrow(data_R2 %>%
                     filter((Title.value == Quanti.title)))/nrow(data_R)*100
taux_exact

#for Quantitative description
data_R2 <- data_R2 %>%
  mutate(Quanti.description = ifelse(sum_doublet_description > 0,
                                     0,
                                     ifelse(sum_troncat_description + sum_hf_description > 0,
                                            0,
                                            Quanti.title)

  ))


#test
data_R2 %>%
  select(Profession, Description.value, Quanti.description) %>%
  filter(Description.value != Quanti.description) %>%
  select(Profession) %>% table()


taux_exact <- nrow(data_R2 %>%
                     filter((Description.value == Quanti.description )))/nrow(data_R)*100
taux_exact

```

```r
#for bias indicator
data_R2 <- data_R2 %>%
  mutate(Bias.indicator = rowMeans(data_R2[c('Quanti.description','Quanti.title')]),
         Bias.indicator = round(Bias.indicator, digits = 4))


data_R2 %>%
  select(Profession, Bias.indicator, Qualitative.result) %>%
  filter(Bias.indicator != Qualitative.result) %>%
  select(Profession) %>% table()


taux_exact <- nrow(data_R2 %>%
                     filter((Bias.indicator == Qualitative.result)))/nrow(data_R)*100
taux_exact
```

(Fig. 28)

Enfin pour notre étude de cas, nous avons décidé d'extraire la hiérarchie des noms de métiers en cherchant le mot 'classe' dans le titre et la description puis normaliser le résultat (fig. 29).

```r
#for classes
data_R2 <- data_R2 %>%
  mutate(class_desc = str_extract(Description, "class(e)?[:blank:][:digit:]+"),
         class_title = str_extract(Title, "class(e)?[:blank:][:digit:]+"))


data_R2 <- data_R2 %>%
  mutate(classe = case_when(stringr::str_extract(class_desc,"[:digit:]+") == stringr::str_extract(class_title,"[:digit:]+")
                            ~ stringr::str_extract(class_desc,"[:digit:]+"),
                            stringr::str_extract(class_desc,"[:digit:]+") != stringr::str_extract(class_title,"[:digit:]+") ~ "ERROR",
                            is.na(class_title) ~ stringr::str_extract(class_desc,"[:digit:]+"),
                            is.na(class_desc) ~ stringr::str_extract(class_title,"[:digit:]+")
                            ))

data_R3 <- data_R2 %>%
  filter(!is.na(classe) & classe != "ERROR") %>%
  group_by(classe) %>%
  mutate(nRow = n(),
         val_bias = 1,
         Qualitative.result = as.character(Qualitative.result)) %>%
  group_by(classe,Qualitative.result,nRow,Profession) %>%
  summarise(val_norm = sum(val_bias)/nRow)
```

(Fig. 29)

Dans le but de tester notre algorithme et d'évaluer notre hypothèse, nous avons réalisé une étude de cas supplémentaire en grattant les intitulés de 150 offres de directeur. Les résultats montrent que le modèle a bien détecté 88.6 % des titres. Les résultats qui n'ont pas été détecté ont été dû aux ponctuations telles que le point médian, la barre oblique et le tiret dans les formes tronquées. Nous avons donc modifié nos expressions régulières pour ensuite avoir un taux d'exactitude de 98 %. Seulement les résultats de 3 titres n'ont pas été bien détecté. Le bruit était causé par la traduction du titre en anglais et l'utilisation du suffixe 'te' (qui ne fait pas partie de nos *Regex* vu qu'il cause plus de bruit en le rajoutant).

4.3. L'exploration de données

Voici la partie la plus importante : l'exploration de données qui comprend la présentation des données, la régression, la corrélation et la prédiction.

a. La présentation des données :

Présenter les données se fait à l'aide des techniques de visualisation des données. Pour visualiser nos données, nous avons eu recours à différentes représentations graphiques qui dépendent du type de données : catégorielles ou quantitatives.

Pour commencer, nous avons visualisé les résultats qualitatifs de nos analyses à l'aide d'un *'Pie-donut chart'* un digramme circulaire pour présenter la part de la parité et l'inégalité du genre dans notre corpus (fig. 30)

```
# plot the quali result with Pie Donut
data_R2 %>%
  select(Profession, quali.bias, Qualitative.result) %>% table()
data_R2 %>%
  PieDonut(aes(Profession, quali.bias))
```

(Fig. 30)

Puis la distribution de l'usage des trois formes d'écriture épicène : l'abréviation H/F, les formes tronquées et les doublets (fig. 31)

```
#plot the epicene writing rules use percentage

data_R2 <- data_R2 %>%
  mutate(trancat = case_when(sum_troncat_description + sum_troncat_title > 0 ~ 1, T ~ 0),
         doublet = case_when(sum_doublet_description + sum_doublet_title > 0 ~ 1 ,T ~0),
         h.f = case_when(sum_hf_description + sum_hf_title > 0 ~ 1,T ~0))

data_R2 %>%
  select(Profession, trancat, doublet, h.f) %>%
  gather(key='variable',value='valeur', trancat, doublet, h.f) %>%
  ggplot(aes(x = Profession,  fill = Profession)) +
  geom_bar(position = "dodge") +
  ggplot2::facet_grid(valeur~variable) +
  geom_text(stat='count', aes(label=..count..), position=position_dodge(width=0.9), vjust=-0.25)
```

(Fig. 31)

Ensuite pour l'étude de cas, nous avons regardé la distribution de classe dans les offres d'emploi à l'aide d'un graphique à barres (fig. 32)

```
#plot classes

data_R3 %>%
  dplyr::mutate(Qualitative.result = case_when(Qualitative.result == "-1" | Qualitative.result == "-0.5" ~ "Biaisé H",
                                               Qualitative.result == "1" | Qualitative.result == "0.5" ~ "Biaisé F",
                                               Qualitative.result == "0" ~ "Non biaisé")) %>%
  ggplot(aes(x = classe, y = val_norm, group=Qualitative.result,fill=Qualitative.result)) +
  geom_bar(stat="identity",position="dodge",width = .5) +
  facet_grid(~Profession)
```

(Fig. 32)

Le graphique à barres montre les comparaisons entre les catégories. Un axe du graphique montre les catégories spécifiques comparées (dans notre cas c'est la variable profession avec

les deux métiers) et l'autre axe représente une valeur mesurée qui est la valeur de biais. Pour avoir un visuel qui explique si nos offres sont biaisées ou pas selon le titre et la description, nous avons fait deux bar plots pour chaque variable (fig. 33).

```
# plot the quanti result with bar plot

# Quanti.description
data_R2 %>%
  mutate(Quanti.description = factor(Quanti.description)) %>%
  ggplot(aes(x = Profession, fill = Quanti.description)) +
  geom_bar(position = "dodge")+
  geom_text(stat='count', aes(label=..count..), position=position_dodge(width=0.9), vjust=-0.25)


#quanti.title
data_R2 %>%
  mutate(Quanti.title = factor(Quanti.title)) %>%
  ggplot(aes(x = Profession, fill = Quanti.title)) +
  geom_bar(position = "dodge")+
  geom_text(stat='count', aes(label=..count..), position=position_dodge(width=0.9), vjust=-0.25)

```

(Fig. 33)

Ensuite, pour comparer les résultats quantitatifs et qualitatifs, nous avons opté pour un boxplot et un violinplot pour les représenter visuellement (fig. 34).

```
#boxplot bias indicator with qualitative result
data_R2 %>%
  mutate(Bias.indicator = factor(Bias.indicator)) %>%
  ggplot(aes(x = Profession, y = Qualitative.result,
             fill = Bias.indicator)) +
  geom_boxplot() +
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=12,face="bold"),
        axis.title.x=element_blank(),
        strip.text.x = element_text(size=12,face="bold"),
        legend.text=element_text(size=12)) +
  scale_fill_viridis_d() +
  geom_point(position = position_jitterdodge(jitter.width = 0.5),
             alpha = 0.3)

#violinplot bias indicator with qualitative result
data_R2 %>%
  mutate(Bias.indicator = factor(Bias.indicator)) %>%
  ggplot(aes(x = Profession, y = Qualitative.result,
             fill = Bias.indicator)) +
  geom_boxplot() +
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=12,face="bold"),
        axis.title.x=element_blank(),
        strip.text.x = element_text(size=12,face="bold"),
        legend.text=element_text(size=12)) +
  scale_fill_viridis_d() +
  geom_point(position = position_jitterdodge(jitter.width = 0.5),
             alpha = 0.3)
```

(Fig. 34)

b. La régression :

Cette méthode nous aide à comprendre s'il y a une relation linéaire entre les variables. (fig. 35).

```
# regression quantitative and qualitative title
data_R2 %>%
  ggplot(aes(x = Title.value, y = Quanti.title, colour = Profession)) +
  geom_point() +
  geom_smooth(method = "lm")


#regression qualitative and quantitative description
data_R2 %>%
  ggplot(aes(x = Description.value, y = Quanti.description, colour = Profession)) +
  geom_point() +
  geom_smooth(method = "lm")


#regression qualitative result & Biais indicator
data_R2 %>%
  ggplot(aes(x = Qualitative.result, y = Bias.indicator, colour = Profession)) +
  geom_point() +
  geom_smooth(method = "lm")

```

(Fig. 35)

c. La corrélation :

Maintenant, nous allons mesurer l'intensité de la corrélation entre les différentes variables en obtenons des coefficients (fig. 36).

```
#correlation
library(GGally)
data_R2 %>%
  select(Quanti.description,
         Quanti.title,
         Bias.indicator,
         Title.value,
         Description.value,
         Qualitative.result) %>%
  ggpairs(lower=list(continuous=wrap("smooth", colour="black")),
          upper = list(continuous = wrap("cor", size=4, colour = "black"))) +
  theme(strip.text = element_text(size = 8),
        axis.text = element_text(size = 9))

```

(Fig. 36)

d. L'arbre de décision :

Cette étape vise à prédire nos résultats qualitatifs à l'aide de l'arbre de décision à l'aide des différentes informations extraites (fig. 37).

```
406
407 #for inference tree
408 #run PCA
409 library(ggfortify)
410 data_R2 %>%
411   select(-c(ID_job, Profession, Title, Description)) %>%
412   select(-contains("tit")) %>%
413   select(-contains("desc")) %>%
414   # run PCA
415   prcomp() %>%
416   autoplot(data_R2 = data_R2,
417            colour = 'Qualitative.result',
418            loadings = TRUE,
419            loadings.colour = "blue",
420            loadings.label = TRUE,
421            loadings.label.colour = "blue")
422
423
424 library(party)
425 ctree(Qualitative.result ~. ,
426       data = data_R2 %>%
427         mutate(Profession = factor(Profession)) %>%
428         mutate(Qualitative.result = factor(Qualitative.result)) %>%
429         select(-c(Title, Description, ID_job)),
430       controls = ctree_control(testtype = "MonteCarlo")) -> Bias_tree
431 plot(Bias_tree)
432
433 # we extract the responses from the tree
434 pred_biais<-predict(Bias_tree)
435 accuracy_table<-table(pred_biais,data_R2$Qualitative.result)
436 accuracy_table
437
```

(Fig. 37)

CHAPITRE III

Les résultats

1. Les résultats

Dans cette partie, nous présentons les résultats de notre étude selon les différentes techniques de *Data Mining.*

a. La visualisation

- **Résultats qualitatifs :**

Fig. 38 : l'écart de genre dans les offres d'emploi selon les résultats qualitatifs

Nous visualisons nos résultats d'analyse manuelle à l'aide d'un diagramme circulaire aussi appelé *donut chart.* D'après le graphique (fig. 38) :

- 56.1 % des offres de secrétaire sont paritaire contre 35.7% d'offres de conducteur paritaires.
- 2 % des offres de secrétaire sont biaisées féminin, cependant, aucune des offres de conducteur n'est biaisée féminin.
- 8.8 % des offres de secrétaire sont moyennement baisées féminin, néanmoins, aucune des offres de conducteur n'est moyennement biaisée féminin.
- 21.6 % des offres de secrétaire sont moyennement biaisées masculin contre 23.8 % d'offres conducteur moyennement biaisées masculin.
- 11.5 % des offres de secrétaire sont biaisées masculin contre 40.5 % d'offres de conducteur biaisées masculin.

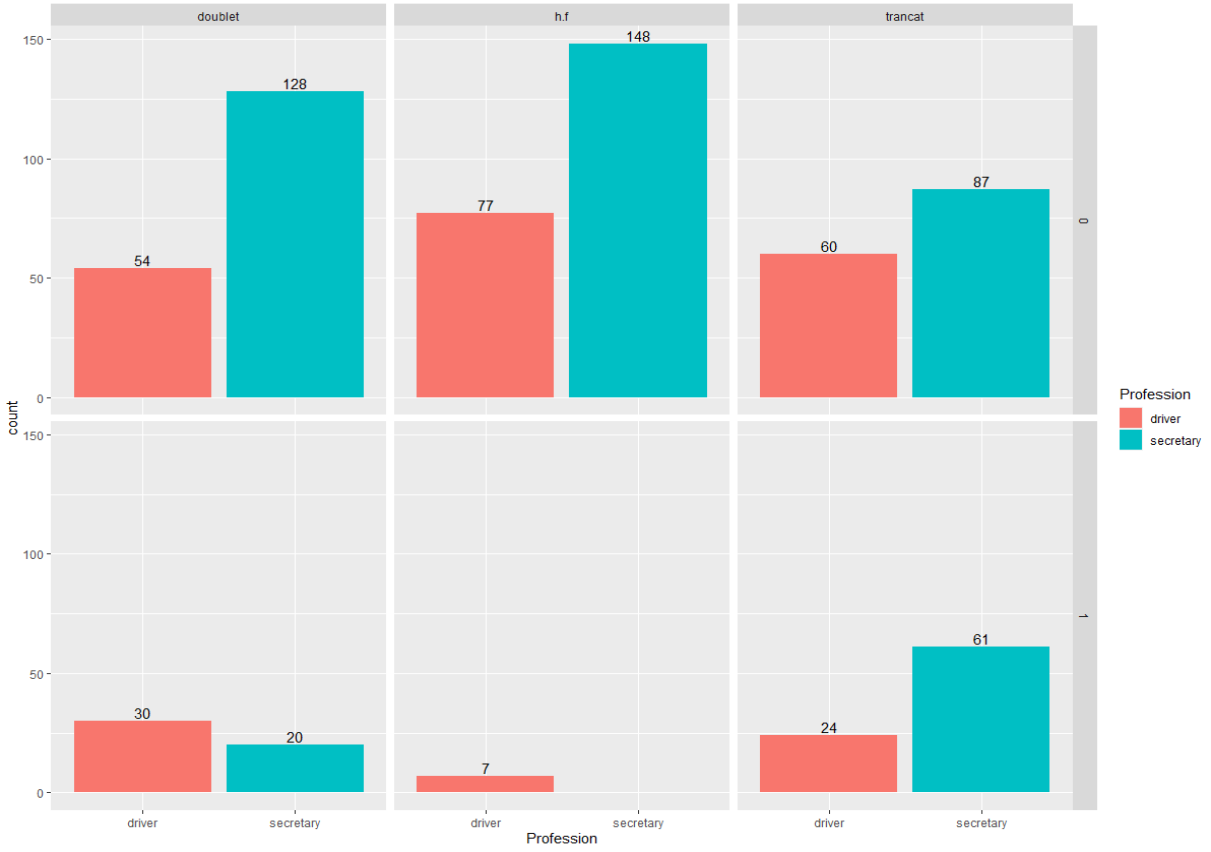


Fig. 39 : la fréquence d'usage des règles d'écriture épicène

Pour avoir une idée sur la fréquence d'usage des trois formes d'écriture épicène (doublet, formes tronqués et abréviation H/F), nous les avons présentées sous un diagramme à bar (fig. 39). En haut du graphique, les offres qui ne comprennent pas une des formes d'écriture

épicène et en bas les offres où la machine a détecté une de ces formes. Nous pouvons ainsi remarquer que :

- Les formes tronquées sont beaucoup plus fréquentes dans le métier de secrétaire que dans le métier conducteur et elles sont généralement les plus utilisées dans les offres d'emploi.
- L'abréviation n'est jamais utilisée pour le métier secrétaire et elle n'est employée que dans 7 offres de conducteur.
- Le doublet est un peu plus employé dans les offres de conducteur que dans les offres de secrétaire.

- **Résultats quantitatifs :**

Pour visualiser nos résultats quantitatifs des deux variables 'Description' et 'Title', nous avons eu recours à un diagramme à bar.

Selon la description (fig. 40) :

- 72.29 % des offres de secrétaire sont paritaires vs 59.52 % d'offres de conducteur paritaires.
- 3.37 % des offres de secrétaire sont biaisées féminin vs 1.19 % d'offre de conducteur biaisées féminin.
- 24.32 % des offres de secrétaire sont biaisées masculin vs 39.28 % d'offres de conducteur biaisées masculin.

La variable 'Description' a un taux de précision de 77.1 %. Ceci dit que 44 offres de secrétaire et 9 offres de conducteur n'ont pas été bien prédites. Comme les résultats de la description repose sur les résultats du titre pour décider le biais du genre, nous pouvons dire que les descriptions biaisées des offres secrétaire ne s'alignent pas avec les titres. Ceci est peut-être dû au fait que secrétaire est un nom de métier épicène[15] et que la description des offres d'emploi de nom de métiers épicène cache plus d'informations sur la parité ou l'inégalité du genre. La défaillance des résultats de la description est dû à la méthode par règles que nous utilisons et

[15] Un mot désignant ou qualifiant un être animé (nom, adjectif, pronom), qui a la même forme au genre masculin et au genre féminin (Arbour et al. 2018).

qui ne permet pas la détection de biais du genre. Un travail de développement de ce modèle augmentera sa performance.

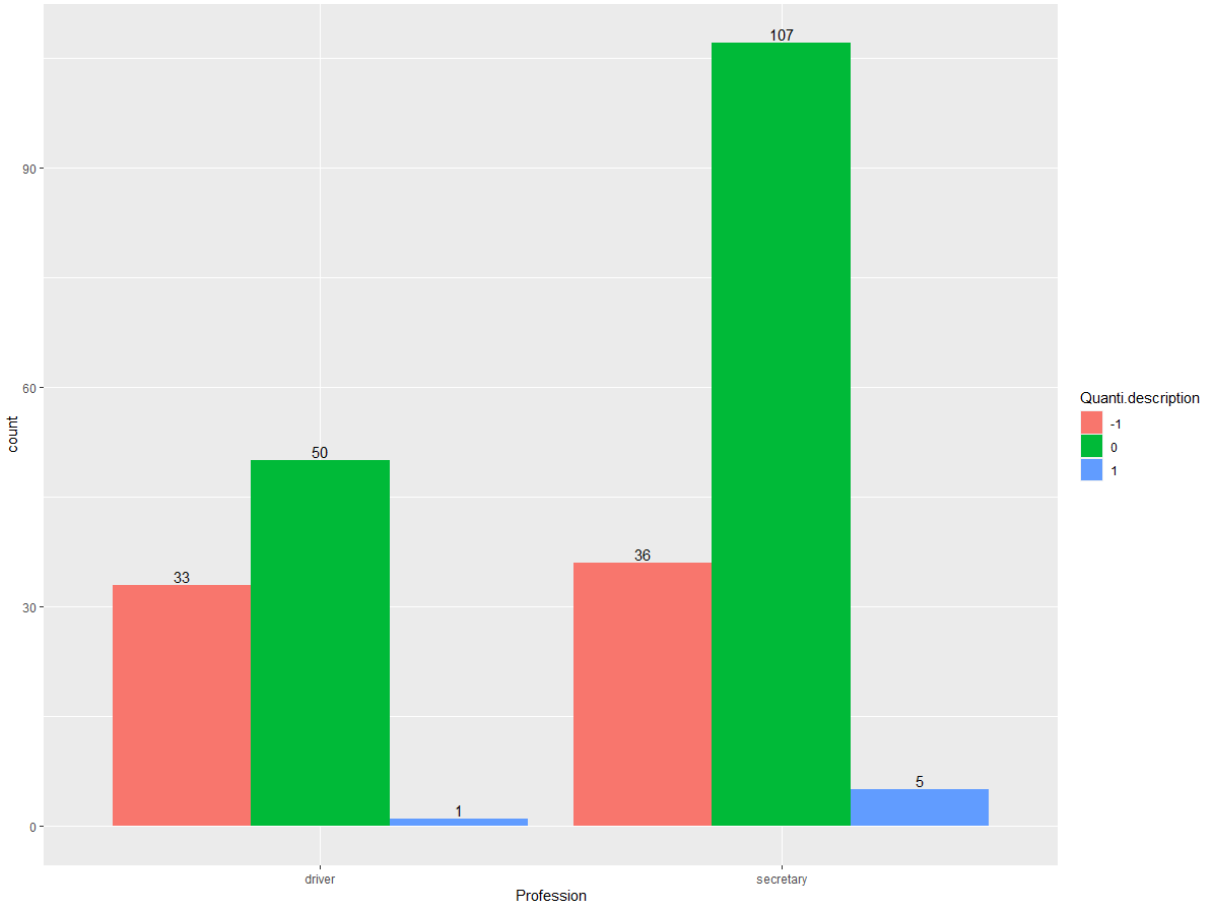


Fig. 40 : l'écart de genre dans les offres d'emploi selon la description

Selon l'indicateur de biais quantitatif de la variable 'Title' (fig. 41) :

- 67.56 % des offres de secrétaire sont paritaires vs 39.28 % d'offres conducteur paritaires.
- 5.40 % des offres de secrétaire sont biaisées féminin vs 1.19 % d'offres de conducteur biaisées féminin.
- 27.02 % des offres de secrétaire sont biaisées masculin vs 59.52 % d'offres de conducteur biaisées masculin.

Selon les résultats de test, 95.6 % des résultats quantitatif de la variable 'Title' sont exacts. Cela veut dire que seulement 10 offres sur 232 offres d'emploi n'ont pas été bien prédites dont 2

offres de secrétaire et 8 offres de conducteur. D'après nos observations, les titres longs ne permettent pas une bonne prédiction à notre algorithme.

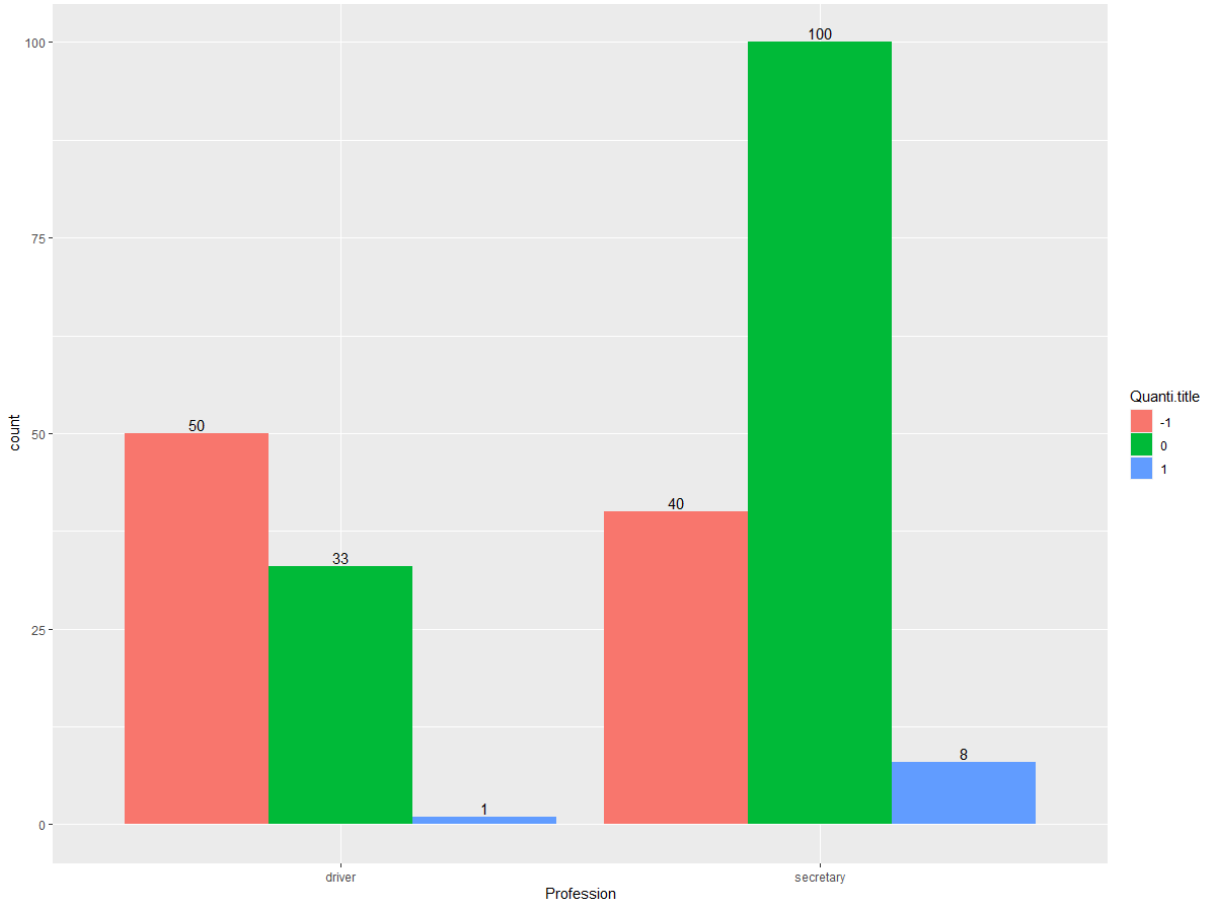


Fig. 41 : l'écart de genre dans les offres d'emploi selon le titre

Nous pouvons donc dire que le titre s'approche de l'annotation manuelle et a de meilleurs résultats comparé à la description. Ceci s'explique par le fait que le titre est limité en taille ce qui rend la méthode par règles efficace en détection de biais et de parité. Cependant la description est longue, les simples *Regex* ne permettent pas la détection de bais seulement la parité (à l'exception des mots et des pronoms épicènes).

Pour comparer les résultats de notre annotation manuelle et celle de la machine, nous avons fait recours à deux graphiques : le boxplot 'boîte à moustaches' (fig. 42) et le violin plot 'tracé de violon' (fig. 43). Notre algorithme a un taux de précision de 75.4 %. D'après les deux graphiques, il prédit plutôt bien les résultats du métier conducteur ainsi que la parité pour le métier secrétaire. Cependant, il prédit mal le biais pour ce dernier. 44 offres d'emploi secrétaire et 13 offres de conducteur n'ont pas été bien prédit.

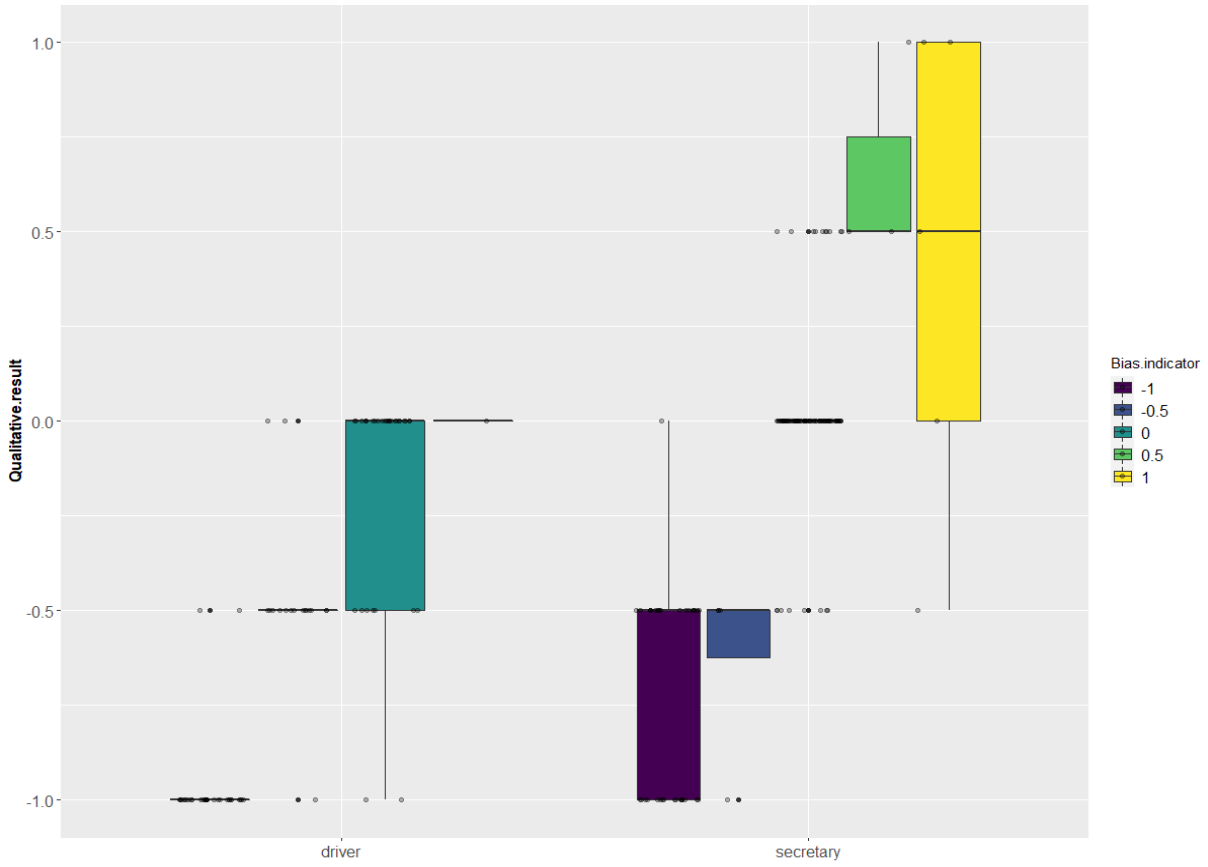


Fig. 42 : comparaison entre les résultats qualitatifs et quantitatifs

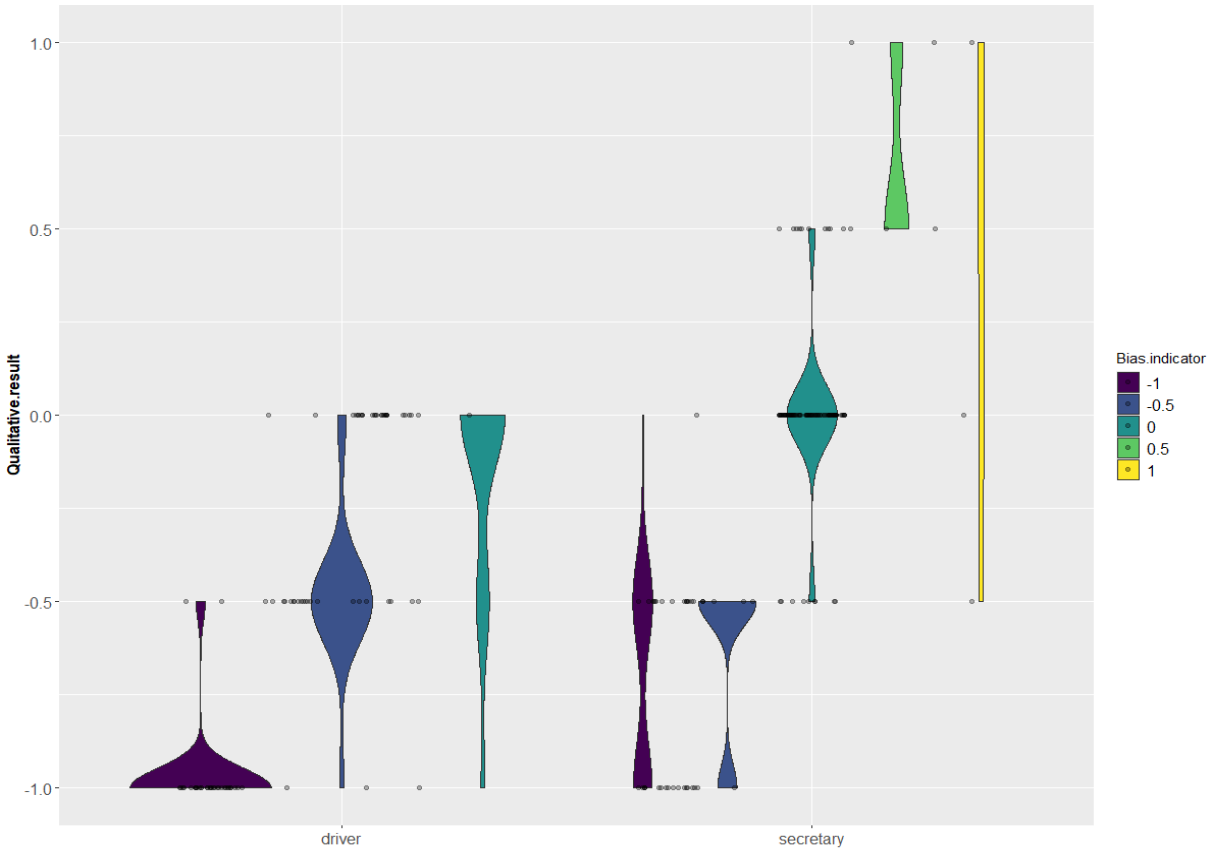


Fig. 43 : comparaison entre les résultats qualitatifs et quantitatifs

a. La régression

Comme déjà mentionné (cf. Régression), la régression détermine le mode d'association entre ses deux variables. Sur la figure (fig. 44), nous avons l'association et la relation linéaire entre les deux variables de l'analyse quantitative et qualitative du 'Title'. Nous pouvons constater une régression linéaire positive entre les deux variables, ils forment quasiment une même ligne droite.

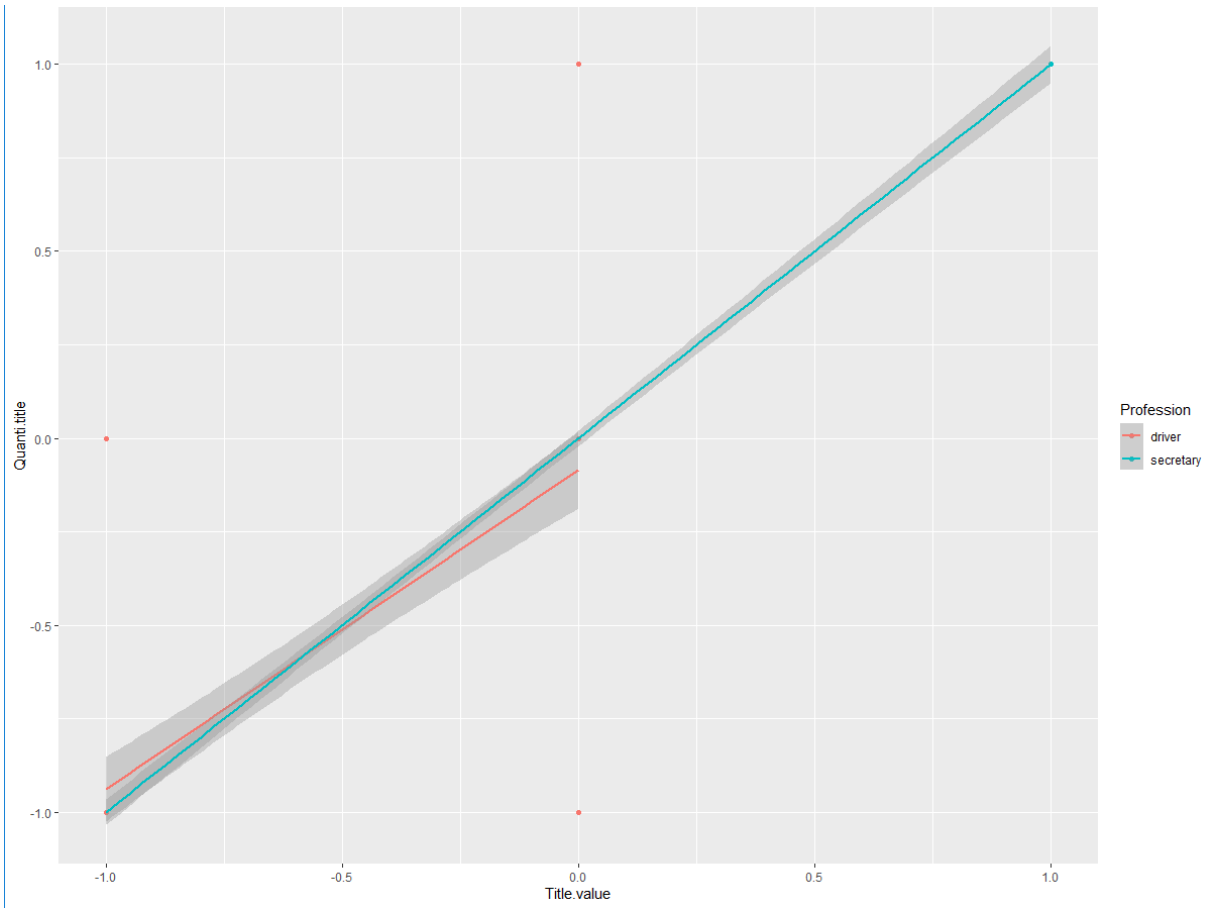


Fig. 44 : le mode d'association entre les variables du titre

Cependant pour la variable 'Description', la corrélation est moins forte (fig. 45).

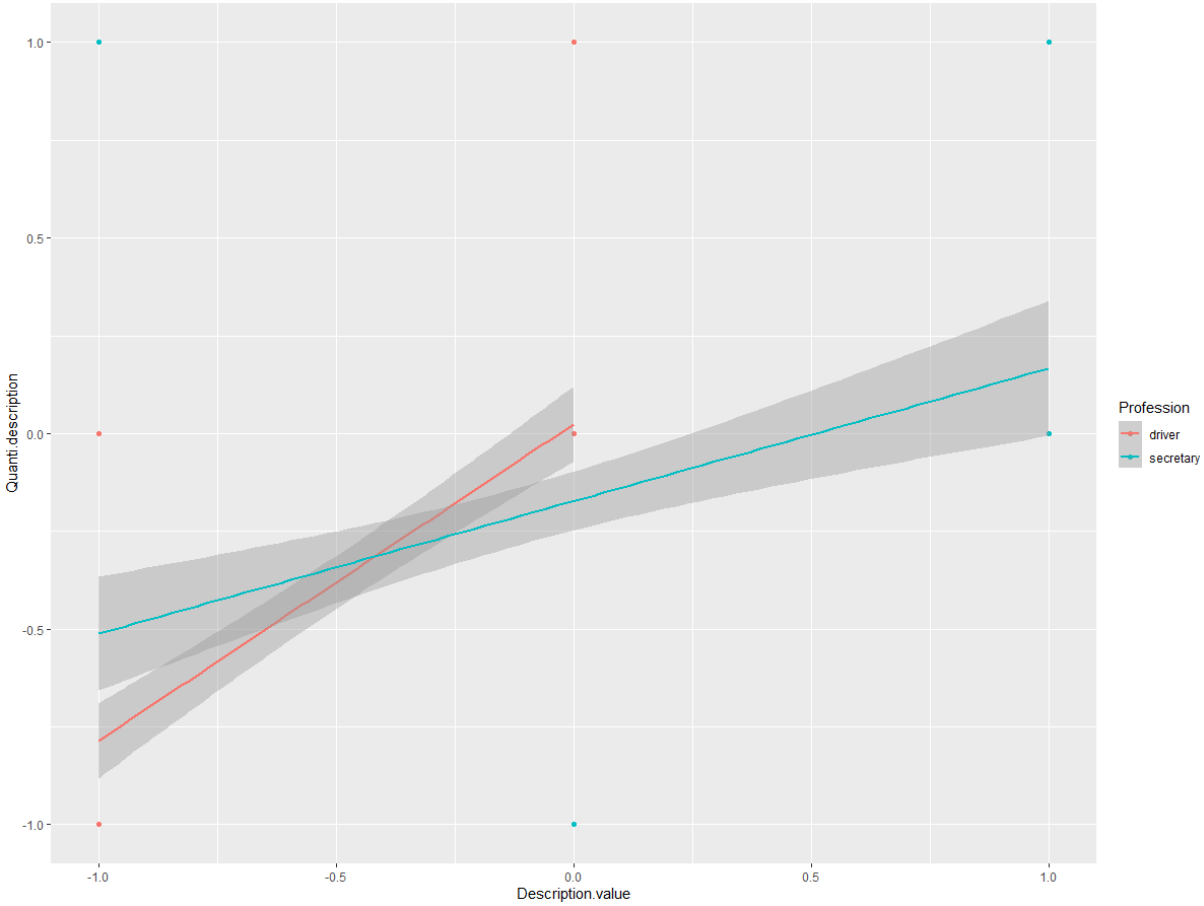


Fig. 45 : le mode d'association entre les variables de la description

Quant à l'indicateur de biais, nous remarquons que plus la valeur de résultat qualitative augmente, plus la valeur de résultat quantitative augmente, cependant quelques points représentant des offres se trouve un peu éparpillés loin de deux lignes. Elles représentent les offres qui n'ont pas été bien prédite par la machine (fig. 46).

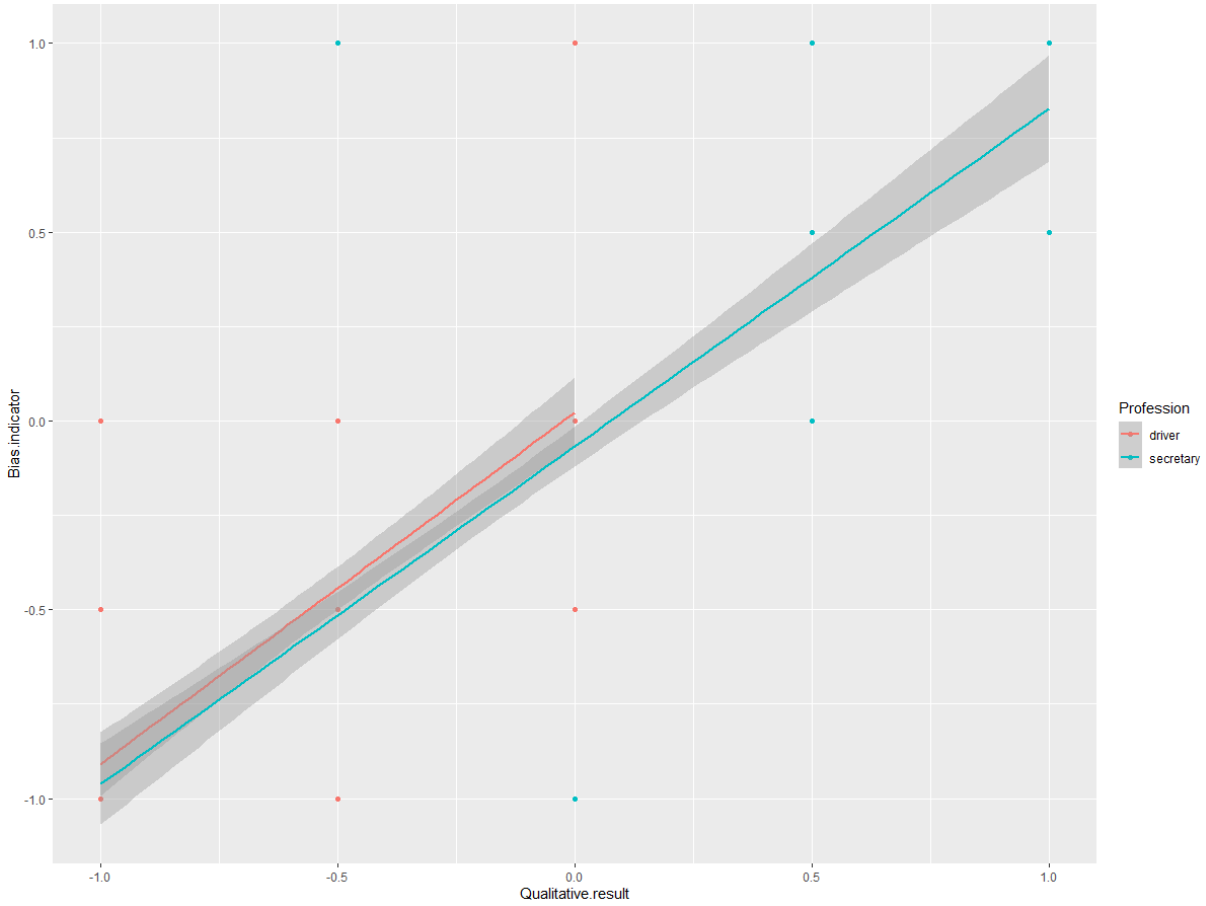


Fig. 46 : le mode d'association entre les variables de l'annotation manuelle et l'algorithme

b. La corrélation

Pour regarder de plus près et mesurer l'intensité de corrélation, nous regardons le corrélogramme (fig. 47). Pour rappel, le coefficient est interprété comme suit :

Coefficient	Corrélation
Près de 0	Nulle
Près de 0.5	Faible
Près de 0.75	Moyenne
Près de 0.87	Forte
Près de 1	Très forte
1	Parfaite

Nous pouvons dire donc que nous avons une corrélation forte avec un coefficient de 0.81 entre notre algorithme et l'annotation manuelle. Quant aux deux variables quantitatives et qualitatives de 'Title', elles corrèlent très fortement avec un coefficient de 0.92. Cependant, les variables quantitatives et qualitatives de 'Description' ont une faible corrélation à modéré avec un coefficient de 0.53.

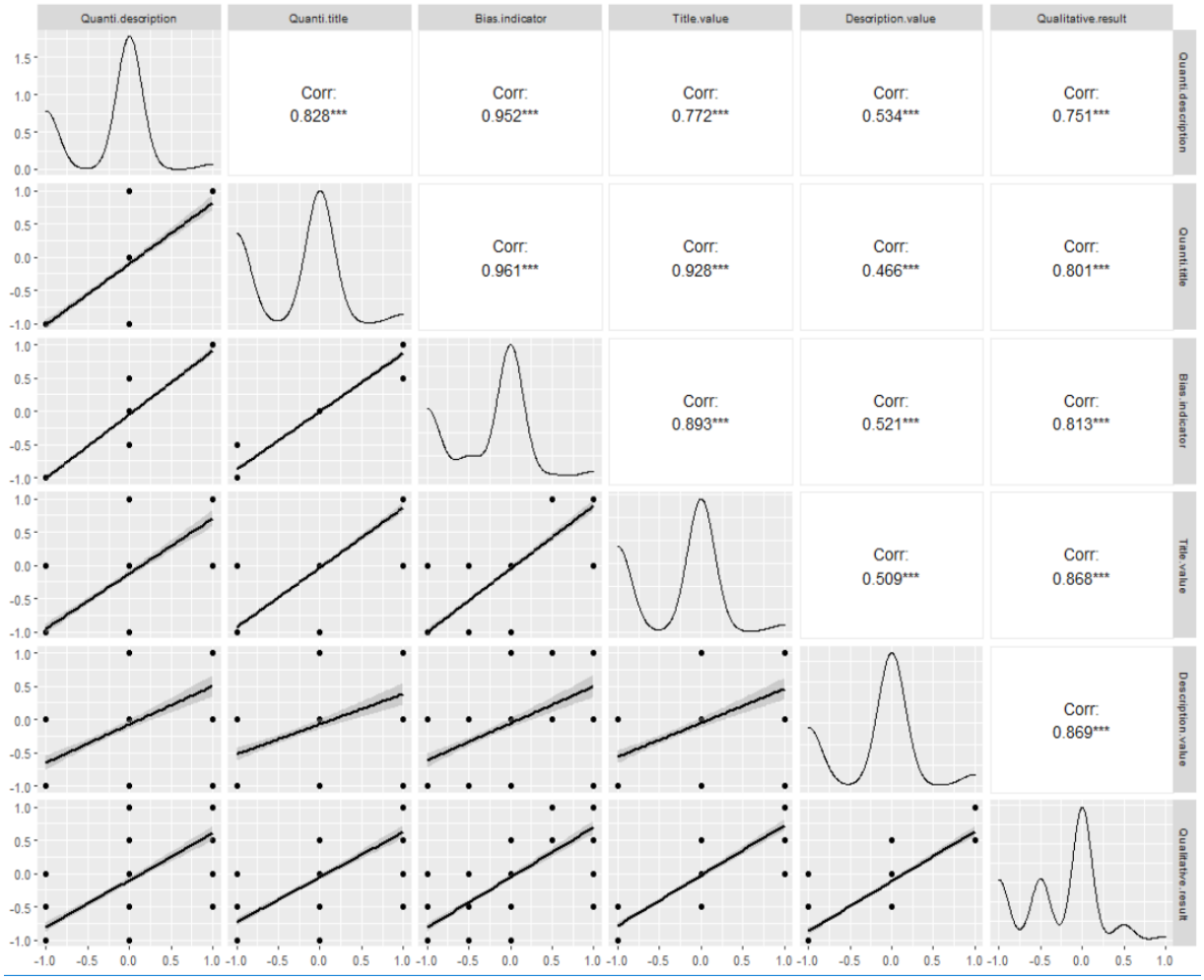


Fig. 47 : mesure de l'intensité de corrélation entre les différentes variables

c. L'arbre de décision

Nous voulons maintenant entrainer la machine à prédire la valeur de résultat qualitatif à partir des informations que nous lui avons données (fig. 48). Pour prédire si une offre d'emploi est biaisée, moyennement biaisée ou paritaire, l'arbre s'est construit sur quatre niveau, nous suivons les numéros en dessus de chaque nœud pour comprendre les décisions :

Si le nœud de racine 'Description value' est :

- Égale ou inférieur à -1 et le nœud 'Title value' est égale ou inférieur à -1, l'offre d'emploi est classé -1 par le nœud de décision dit *leaf node*.
- Égale ou inférieur -1 et le nœud 'Title value' est supérieur à -1, l'offres est classé -0.5 et 0.

- Supérieur à -1 et le nœud ‘Title value’ est égale ou inférieur à -1, le nœud de décision donne -0.5.
- Supérieur à -1 et le nœud ‘Title value’ est supérieur à -1, l’algorithme regarde encore le nœud ‘Description value’ s’il est inférieur ou égale à 0, il décidera de classer l’offre comme 0 ou 0.5 mais s’il est supérieur à 0, il donnera une valeur de 0.5 ou 1.

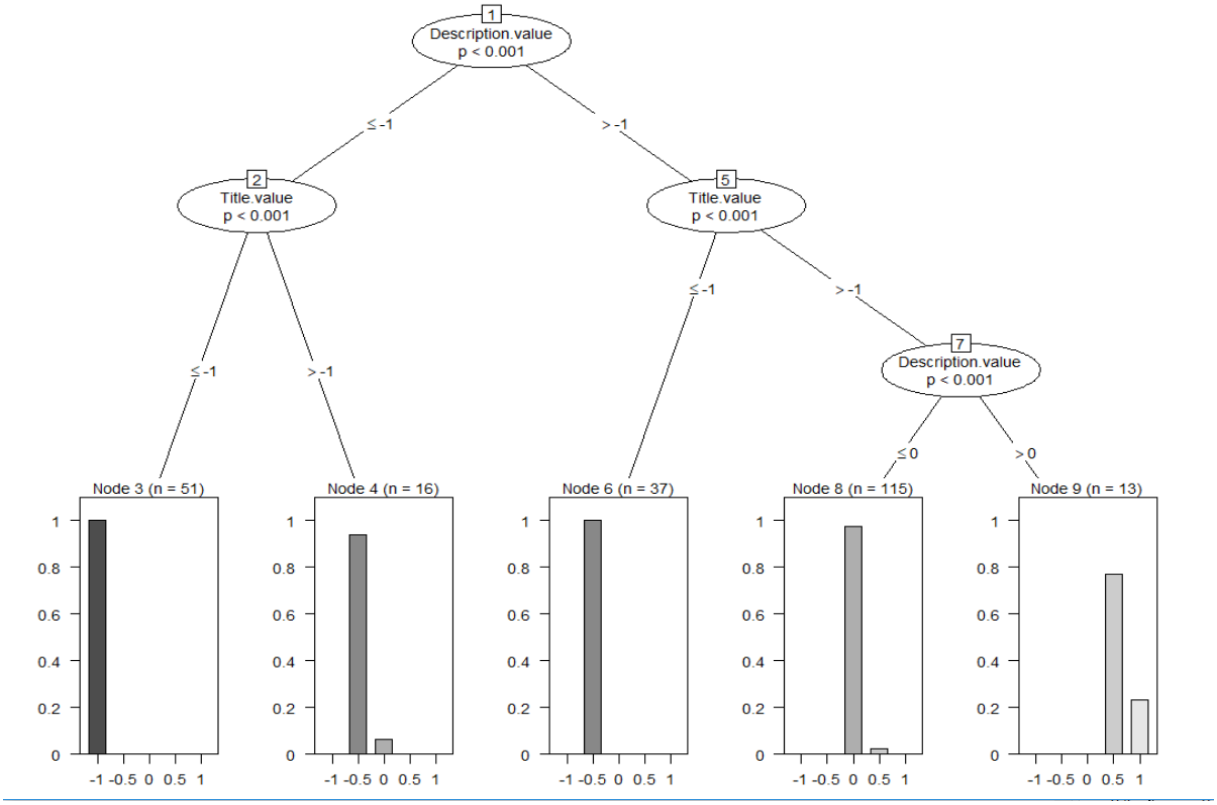


Fig. 48 : arbre de prédiction des résultats qualitatif

Nous générons ensuite un tableau croisé (fig. 49) entre les valeurs réelles et les prédictions afin d’estimer la précision de notre modèle. Nous pouvons constater que le biais masculin de valeur -1 et -0.5 est bien prédit sans erreur. Pour la parité avec la valeur 0, l’algorithme a réussi à prédire toutes les offres sauf une à laquelle a donné une valeur de -0.5. Pour les offres moyennement biaisées féminin, notre arbre les a bien classés sous 0.5 sauf 3 offres auxquelles elle a donné une valeur de 0. Finalement le biais féminin n’a pas été prédit, trois offres biaisées féminin ont été classé comme moyennement biaisées féminin.

pred_biais	-1	-0.5	0	0.5	1
-1	51	0	0	0	0
-0.5	0	52	1	0	0
0	0	0	112	3	0
0.5	0	0	0	10	3
1	0	0	0	0	0

Fig. 49 : tableau croisé d'estimation de précision du modèle

2. La discussion et interprétation

D'après nos résultats quantitatifs, les recruteurs font plus recours aux formes tronquées aussi dit doublets abrégés pour exprimer la parité dans leurs annonces même si l'office québécois de la langue française ne recommande pas leur emploi sauf dans le cas d'un espace restreins. Ceci est peut-être dû au fait que les formes tronquées sont plus rapides à écrire comparées aux doublets. Le doublet vient en deuxième position et enfin l'abréviation H/F qui a été très peu présente dans notre corpus bien qu'Indeed la propose.

La parité est plus forte dans les offres de secrétaire que dans les offres de conducteur. Une hypothèse pourrait dire que les recruteurs imaginent moins les femmes faisant des métiers dit masculin mais plus facilement des hommes faisant des métiers dit féminins. Une autre explication serait que lors du grattage de données, nous avons récolté des offres proches de secrétaire comme adjoint administratif qui peut être un métier plus paritaire.

Selon nos graphiques présentant nos résultats qualitatifs, le métier conducteur n'est jamais biaisé ou moyennement biaisé féminin. Cependant il est fortement biaisé masculin avec un pourcentage de 64.3 % d'annonces biaisées ou moyennent biaisées masculin. Ceci rejoint notre hypothèse de départ qui dit que les offres d'emploi sont biaisées suivant le stéréotype. Le métier de conducteur est stéréotypé masculin avec une très forte présence masculine en plus des offres d'emploi biaisé homme.

Le métier de secrétaire quant à lui, est stéréotypé féminin mais il est plus biaisé masculin (33.1 % de biais masculin) que féminin (10.8% de biais féminin) d'après notre corpus. Ceci peut être dû aux échelons et la hiérarchie. Les femmes sont sous-représentées à tous les échelons sauf le premier échelon où elles représentent 50% des employés. Plus l'échelon est élevé moins les femmes sont représentés (Mckinsey, 2019). Nous sommes donc allés explorer cette piste.

3. Etude de cas

Pour comprendre le biais derrière le métier de secrétaire, nous sommes allés explorer nos données. 78 offres comprennent la mention de classe dont 40 offres de secrétaire. D'après le graphique (fig. 50), nous pouvons déjà remarquer que :

- Le biais masculin est présent dans les cinq premières classes hors que le biais féminin n'est présent que dans la première classe avec une seule offre et la neuvième classe avec une seule offre moyennement biaisé.
- La première et la quatrième classe ont une fréquence d'offres biaisé masculin plus élevée et la classe 5 ne comprend que des offres biaisées masculin.

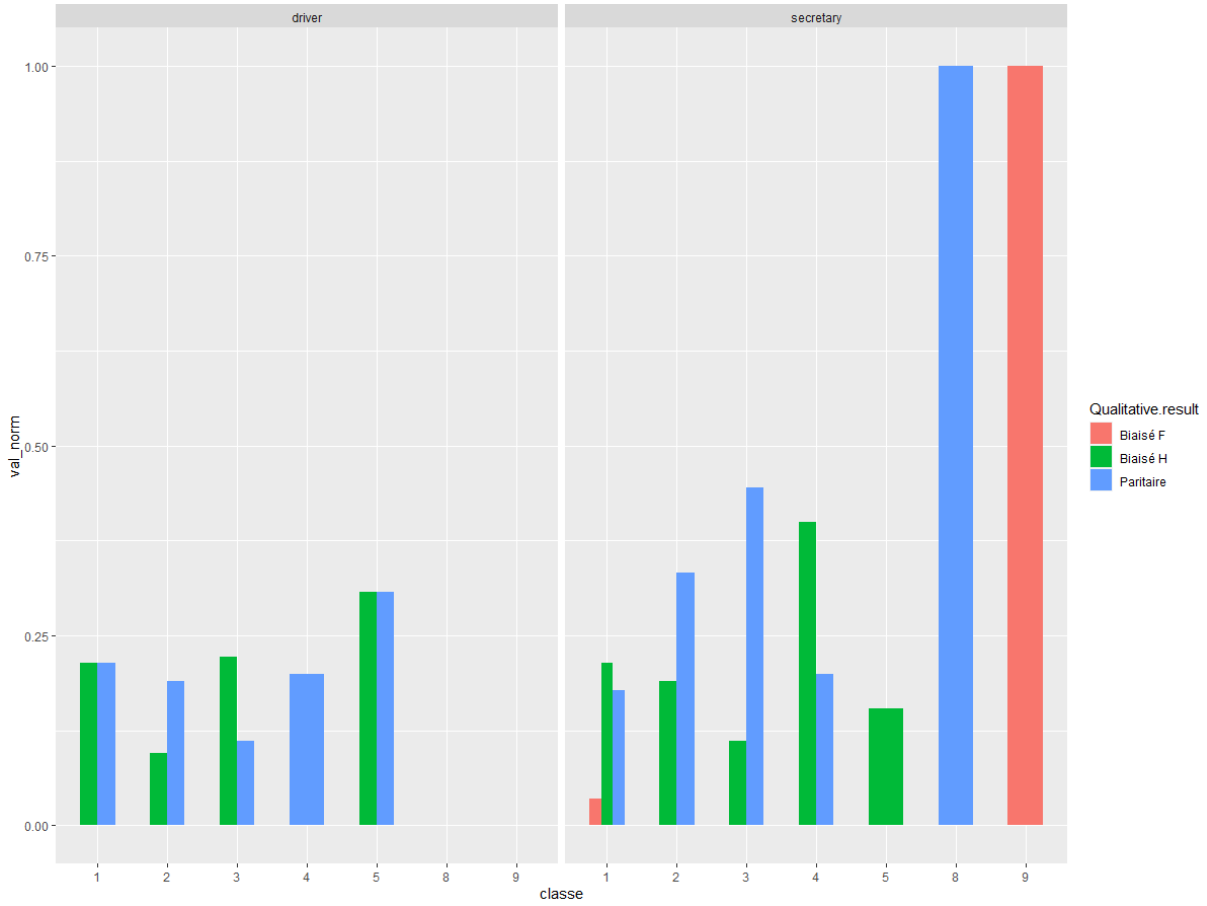


Fig. 50 : distribution des offres selon les classes

Afin de tester notre postulat, nous avons analysé 150 titres d'offre d'emploi 'directeur' grattées depuis le site Indeed Canada. D'après les résultats (fig. 51), 52 % des titres des offres d'emploi sont paritaires vs 48 % biaisés masculin. Le même constat se répète, les offres à échelon élevé sont fortement biaisées masculin ou paritaires mais pas biaisées féminin.

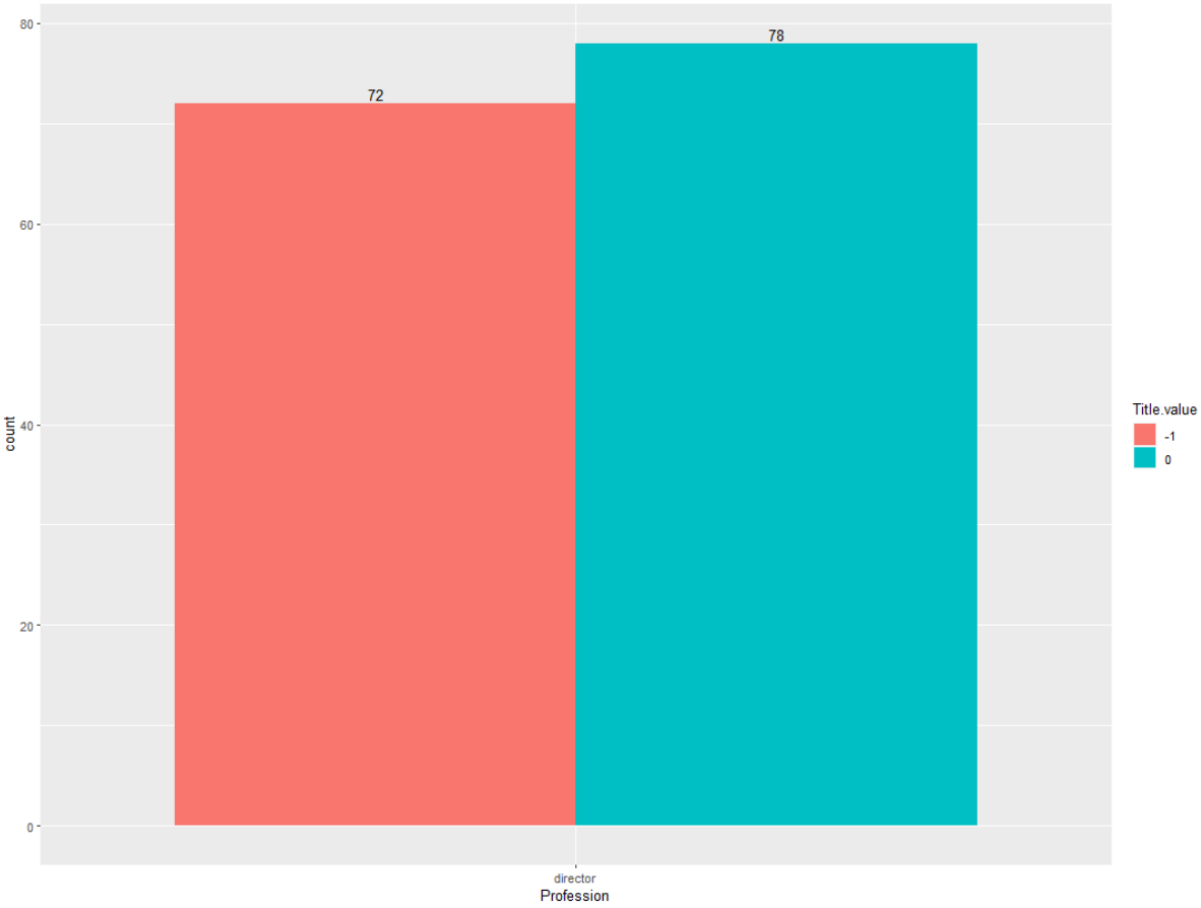


Fig. 51 : l'écart de genre dans les offres d'emploi de 'directeur' selon le titre

4. Les approfondissements possibles

Notre étude est certainement limitée par plusieurs facteurs. D'abord, la limite du temps, nous avions seulement six mois pour construire notre projet de stage, acquérir de nouvelles compétences, les appliquer ensuite et enfin écrire le rapport de stage long. La limite de nos techniques : notre approche quantitative s'est basée sur les expressions régulières considérées comme limitées dans notre cas, une alternative aurait été de faire recours à d'autres techniques de *Text Mining* tels que la lemmatisation. Finalement, la limite du corpus, un corpus plus large aurait peut-être témoigné de résultats plus précis. Malgré toutes ces limites, nous avons pu mesurer l'inégalité du genre et créer un modèle qui détecte le biais du genre, et qui pourra être amélioré dans le futur.

CONCLUSION GENERALE

CONCLUSION GENERALE

Tout au long de ce rapport, nous avons essayé de mesurer et de comprendre l'inégalité du genre dans les offres d'emploi d'Indeed Canada. Nous nous sommes interrogés : comment s'adresser aux genres en recrutement ? Comment s'adresse les recruteurs aux hommes et aux femmes dans leurs annonces ? Les offres d'emploi sont-elles paritaires? Puis nous avons proposé l'hypothèse suivante : les offres d'emploi pourraient être biaisées suivant les stéréotypes.

A travers le premier chapitre, et dans une première section, nous avons présenté notre établissement d'accueil, ses différents axes ainsi que les différentes tâches effectuées. Quant à la deuxième section, nous l'avions dédié au projet EVOGRAM, aux techniques utilisées ainsi qu'à la question de la langue française et de la féminisation linguistique. Enfin, nous avons réservé un chapitre à la présentation des données, la méthodologie choisie, et les différents résultats.

Nos résultats qualitatifs ont montré qu'effectivement plus de la moitié des offres d'emploi de la profession conducteur ont été biaisées homme suivant le stéréotype du métier, cependant la profession secrétaire, et contrairement à ce que dit notre hypothèse, a présenté plus de biais masculin que féminin. D'après notre cas d'étude, les offres à échelon plus élevé ont marqué un fort biais masculin mais jamais féminin.

D'après ce qui a été dit nous pouvons confirmer, en partie, notre hypothèse en disant que les offres d'emploi sont biaisées suivant les stéréotypes sauf dans le cas de postes à haute responsabilité où il peut éventuellement avoir un fort biais masculin. Ceci concorde avec les recherches existantes, en prenant comme exemple Missoffe (2015) :

> Dans la sphère professionnelle, les personnages féminins sont assignés à certaines catégories socioprofessionnelles, généralement socialement sous-considérées. Habileté et concentration sont des qualités professionnelles dites féminines, là où la force physique et morale, l'autorité et l'autonomie sont des caractéristiques dites masculines par excellence.

Cette étude n'est qu'un point de départ pour une étude ultérieure plus approfondie permettant la création d'un modèle de détection et mesure de biais de genre plus puissant et une étude expérimentale sur l'effet de l'écriture épicène sur les candidats dans les offres d'emploi.

Pour conclure, nous aimerions souligner que l'emploi du masculin générique défendu par la grammaire traditionnelles et prescriptives peut avoir des conséquences sur la volonté des femmes à postuler. Le genre grammatical est un élément déclencheur de biais en faveur des hommes (Gygax et al, 2008 ; Gygax et al, 2019) (Houdebine-Gravaud, 1995) et contribue au croyances les représentations mentales (Richy & Burnett, 2021).

L'équilibration du texte épicène en utilisant les différents procédés et règles d'écriture épicène (cf. 5.2. Les règles de l'écriture épicène) est très importante non seulement pour assurer la bonne compréhension mais aussi pour éviter les représentations mentales. Selon l'étude expérimentale mené par Richy & Burnett en 2021, le biais masculin s'étend aussi à l'épicène. L'épicène féminin ne déclenche aucun biais, néanmoins, l'épicène masculin déclenche une représentation masculine.

D'un point de vue technologique, l'IA n'a certainement pas de biais cognitif, cependant, les algorithmes et les modèles de langue présentent des biais importants (Névéol et al, 2022). Ils sont nourris par des données biaisées ce qui ne met pas les modèles à l'abris des stéréotypes genrés dont les algorithmes affinitaires de recrutement d'où vient l'importance d'un travail approfondi sur la question.

BIBLIOGRAPHIE

BIBLIOGRAPHIE

Andrea, E. (2020). *Égalité des genres.* L'encyclopédie canadienne [en ligne]. Consulté le 25 août 2022 sur : https://www.thecanadianencyclopedia.ca/fr/article/egalite-des-sexes#:~:text=L'%C3%A9galit%C3%A9%20des%20genres%20est,les%20m%C3%AAmes%20droits%20et%20privil%C3%A8ges .

Ange, R., Gilles, B., & François, P. (2022). GenderedNews: Une approche computationnelle des écarts de représentation des genres dans la presse française. *ArXiv*. Consulté le 20 juillet 2022 sur : https://doi.org/10.48550/arXiv.2202.05682

Appelbaum, S. H., & Emadi-Mahabadi, S. (2022). Gender Parity in The Workplace: How COVID-19 Has Affected Women. European Journal of Business and Management Research, 7(1), 1–8. Consulté le 30 juillet 2022 sur: https://doi.org/10.24018/ejbmr.2022.7.1.1169 .

Arbour, M., de Nayves, H. & Royer, A. (2014). Féminisation linguistique : étude comparative de l'implantation de variantes féminines marquées au Canada et en Europe. *Langage et société*, 148, 31-51. Consulté le 20 juillet 2022 sur : https://doi.org/10.3917/ls.148.0031

Arbour, M.-È., & De Nayves, H. (2018). Formation sur la rédaction épicène, [En ligne], Office québécois de la langue française. Consulté le 15 février 2022 sur : https://www.oqlf.gouv.qc.ca/redaction-epicene/20180112_formation-redaction-epicene.pdf

Becquer, A., Cerquiglini, B., & Cholewka, N. (1999). *Femme, j'écris ton nom… : guide d'aide à la féminisation des noms de métiers, titres, grades et fonctions.* La Documentation Française. Consulté le 22 juillet 2022 sur : https://www.vie-publique.fr/sites/default/files/rapport/pdf/994001174.pdf

Brenda, C. (1993). Définition et mesure de l'équité en matière d'emploi. *L'emploi et le revenu en perspective*, 5(4). Consulté le 15 juin 2022 sur : https://www150.statcan.gc.ca/n1/fr/pub/75-001-x/1993004/article/38-fra.pdf?st=eXTyikar

Gouvernement de Canada. (2022). *Égalité et inclusion dans les industries et milieux de travail sous réglementation fédérale.* Consulté le 01 juin 2022 sur : https://www.canada.ca/fr/services/emplois/milieu-travail/droits-personne.html

John, L. T. (2017). *R for excel users: an introduction to r for excel analysts.*

Lionel, D. (2016). Analyse qualitative du contenu des représentations sociales. *Les représentations sociales,* pp 85-102. Consulté le 20 juin 2022 sur : https://hal-amu.archives-ouvertes.fr/hal-01648424/document

Misersky, J., Gygax, P., Canal, P., Gabriel, U., Garnham, A., Braun, F., et al. (2014). Norms on the gender perception of role names in Czech, English, French, German, Italian, Norwegian, and Slovak. *Behav. Res. Meth.* 46, 841–871. Consulté le 22 août 2022 sur : https://doi.org/10.3758/s13428-013-0409-z

Missoffe, P. (2015). Stéréotypes, représentations sexuées et inégalités de genre dans les manuels scolaires. *La Revue des droits de l'homme [En ligne].* consulté le 30 août 2022 sur : http://journals.openedition.org/revdh/1667

Névéol, A., Dupont, Y., Bezançon, J., & Fort, K. (2022). French CrowS-Pairs: Extension à une langue autre que l'anglais d'un corpus de mesure des biais sociétaux dans les modèles de langue masqués. HAL-Inria. Consulté le 23 août 2022 sur : https://hal.inria.fr/hal-03680574/document

Regular expressions. Consulté le 20 juillet 2022 sur: https://stringr.tidyverse.org/articles/regular-expressions.html

Richy, C., & Burnett, H. (2021). Démêler les effets des stéréotypes et le genre grammatical dans le biais masculin : une approche expérimentale. *GLAD!.* Consulté le 25 août 2022 sur : https://doi.org/10.4000/glad.2839

Robert, A. & Bouillaguet, A. (2007). *L'analyse de contenu*. Presses Universitaires de France. Consulté le 03 juin 2022 sur : https://doi.org/10.3917/puf.rober.2007.01

Romelaer, P. (2005). Chapitre 4. L'entretien de recherche. Dans : , P. Roussel & F. Wacheux (Dir), Management des ressources humaines: Méthodes de recherche en sciences humaines et sociales (pp. 101-137). Louvain-la-Neuve: De Boeck Supérieur. Consulté le 25 août 2022 sur : https://doi.org/10.3917/dbu.rouss.2005.01.0101

Sandrine, D., Geneviève, B., Anu, M., Mekala, K., Tina, P., Han, Z., & Marissa, N. (2019). *Woman Matter : État des lieux et avenir des femmes sur le marché du travail canadien.* McKinsey & Compagnie. Consulté le 03 juin 2022 sur : https://www.mckinsey.com/~/media/mckinsey/featured%20insights/gender%20equality/the%20present%20and%20future%20of%20women%20at%20work%20in%20canada/20190602-women-matter-2019-vf.pdf

Simon, G. K. (2020). *Écarts de rémunération femmes-hommes : surtout l'effet du temps de travail et de l'emploi occupé* (publication n° 1803). Insee. Consulté le 03 juin 2022 sur : https://www.insee.fr/fr/statistiques/4514861

The R Foundation. *R documentation.* Consulté le 30 juin 2022 sur : https://www.rdocumentation.org

The R Foundation. *What is R?*. R. Consulté le 30 juin 2022 sur : https://www.r-project.org/about.html

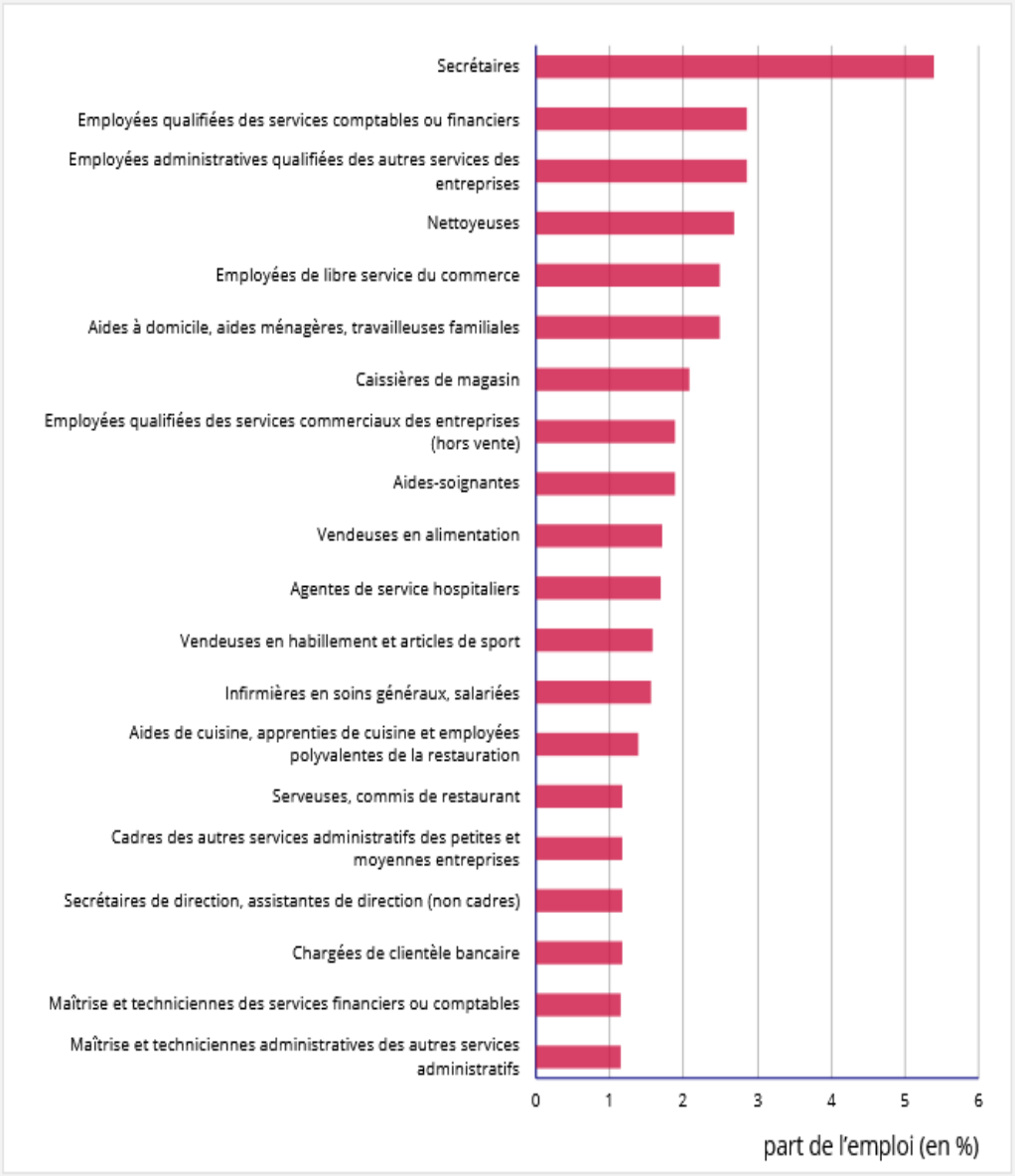
Vachon-l'heureux, P., Guénette, L. (2006). *Avoir bon genre à l'écrit : guide de rédaction épicène.* Office québécois de la langue française. Les Publications du Québec.

ANNEXE

◉ Chez les femmes ○ Chez les hommes

GRAPHIQUE TABLEAU

Figure 3a - Les vingt professions les plus fréquentes chez les femmes



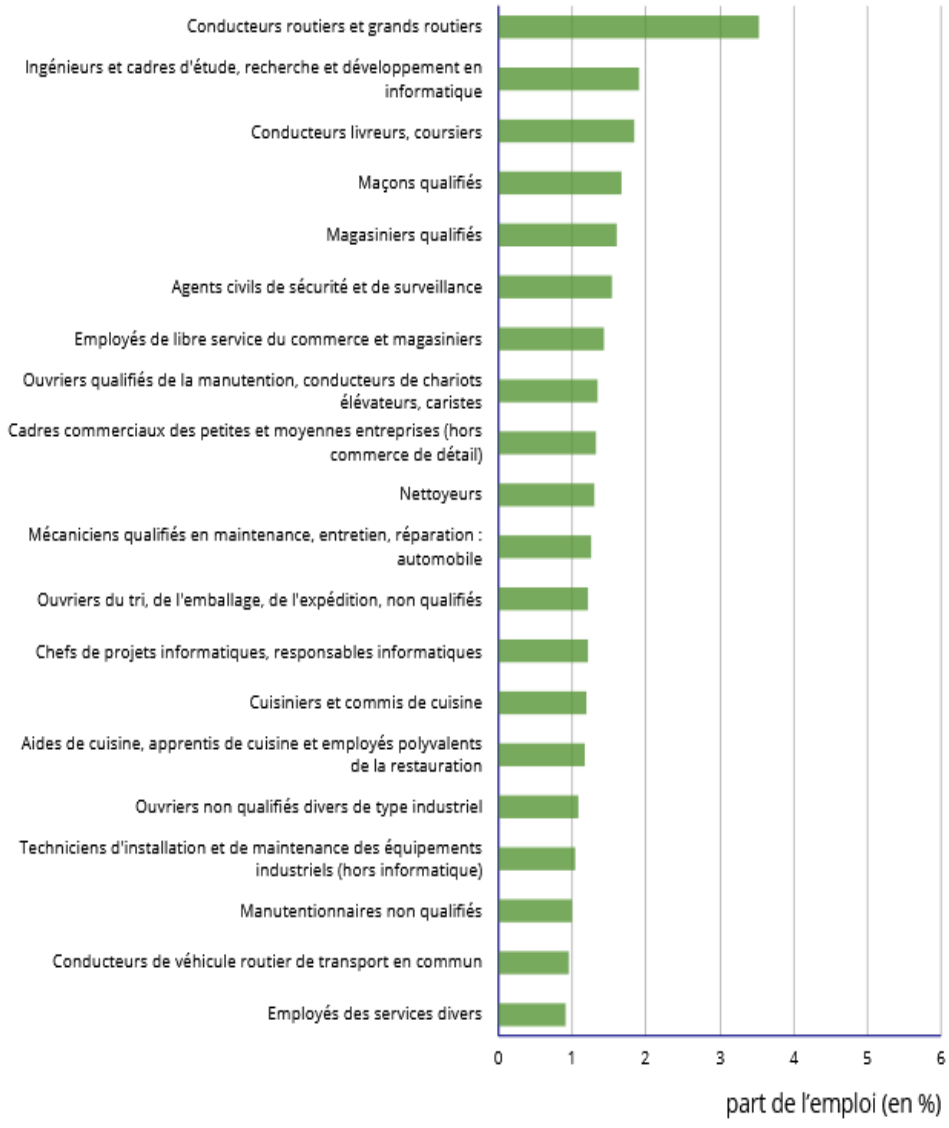
Champ : postes du secteur privé, France métropolitaine, hors apprentis et stagiaires, hors agriculture, hors salariés des particuliers.

Source : Insee, déclarations annuelles de données sociales (DADS) et déclarations sociales nominatives (DSN), 2017.

Chez les femmes **Chez les hommes**

GRAPHIQUE TABLEAU

Figure 3b - Les vingt professions les plus fréquentes chez les hommes



Champ : postes du secteur privé, France métropolitaine, hors apprentis et stagiaires, hors agriculture, hors salariés des particuliers.

Source : Insee, déclarations annuelles de données sociales (DADS) et déclarations sociales nominatives (DSN), 2017.


SCAN ME